# 생존성 강화를 위한 침입감내 시스템의 분류와 통합 프레임워크 제안

김 기 한[†] · 최 명 렬[††] · 이 경 환[†††]


## 요 약

기존의 보안관점은 악의적인 사용자의 공격이나 우발적인 사고로부터 프로그램과 데이터의 보호를 강조한다. 이를 위한 방화벽, 침입탐지 시스템의 연구개발은 성숙 단계에 접어들었다. 최근들어 새롭게 대두되고 있는 침입감내 개념은 정보 생존성을 위한 마지막 방어선으로 공격이 성공하더라도 임무 수행에 필수적인 시스템의 중요 서비스를 계속 제공하기 위해 가용성과 무결성을 강조하는 개념이다. 본 논문에서는 침입감내 시스템을 프로그램 보호와 데이터 보호관점에서 분류하고 프로그램과 데이터의 침입감내 특성을 모두 지원하는 통합된 침입감내 프레임워크를 제안한다.


# Classification of the Intrusion Tolerant Systems and Integrated Framework for Survivability Enhancement

Gi Han Kim[†] · Myeong Ryeoi Chio[††] · Kyung Whan Lee[†††]


## ABSTRACT

Currently security researchers focus on protection of program and data from malicious users and accidents. Therefore, many firewalls and intrusion detection systems have been developed commercially. The intrusion tolerance is a new concept that is the last line of defense for the information survivability. It emphasizes availability and integrity to provide critical system services continuously even when system is compromised. In this paper, we classify current intrusion tolerant technologies from the point of view of program and data. Furthermore, we propose an integrated framework that supports intrusion tolerance of program and data.


**키워드 : 침입감내 시스템(Intrusion Tolerant System), 정보보증(Information Assurance), 정보생존성(Information Survivability)**

## 1. 서 론

1세대 보안 메커니즘은 신뢰할 수 있는 컴퓨팅 환경을 제공하기 위해 암호화, 인증 등을 이용하여 기밀성과 접근 제어 기능을 제공하고, 2세대 보안 메커니즘은 경계 제어, 침입탐지 시스템, PKI, 생체인식 기술로서 침입에 대한 탐지 기능을 제공하였다. 이러한 1세대, 2세대 보안 메커니즘은 공격에 대한 방어와 탐지 기능은 제공하지만 시스템의 생존성과 저항성 기능을 제공하지 않는다. 생존성과 저항성 기능을 위해서는 새로운 보안 매커니즘을 필요로 한다[1].

본 논문은 침입이 성공하였다고 하더라도 시스템의 중요 서비스를 지속적으로 제공하는 것을 목표로, 무결성과 가용성을 강조하는 침입감내 시스템을 프로그램과 데이터 관점, 복제기반(replica-based) 침입감내와 계층기반(layer-based) 침입감내 관점에서 분류하고, 네 가지 침입감내 관점을 모두 지원할 수 있는 통합 침입감내 프레임워크를 제시한다.

4가지 분류의 기준에서 프로그램과 데이터는 보호대상에 따른 분류이고 계층기반과 복제 기반의 분류는 기술적인 접근법에 따른 분류로서 이들의 분류 기준이 서로 다르지만 4가지 분류에 해당하는 시스템은 각각 그 기능성과 적용가능한 환경, 구현 기술이 상이하므로 침입감내 시스템을 4가지로 분류하여 설명한다.

현재 진행 중인 침입감내 시스템의 분류를 통해 침입감내 프레임워크의 전체적인 구성요소를 식별하고, 각 구성요소에서 제공해야 하는 기능을 정의하고, 정의된 기능을 통합한 프레임워크를 서버에 적용한다면 서버의 공격 저항성은 향상될 것이다.

본 논문은 2장에서 DARPA에서 진행되어온 침입감내 시스템 관련 연구를 소개하고, 3장에서 복제기반과 계층기반의 침입탐지 시스템을 프로그램과 데이터관점에서 분류하고, 4장에서 현재 연구가 진행중인 침입감내 시스템 사례연구를 알아보고, 5장에서는 복제기반과 계층기반이 통합된 침입감내 프레임워크를 제시하고, 6장에서는 본 논문이 제

† 종신회원 : ETRI 부설 국가보안기술연구소 연구원
†† 정 회 원 : ETRI 부설 국가보안기술연구소 선임연구원
††† 정 회 원 : 중앙대학교 컴퓨터공학과 교수
논문접수 : 2002년 9월 23일, 심사완료 : 2003년 4월 8일

시한 침입감내 프레임워크에 대한 평가를 수행하고, 7장에서 결론을 맺는다.

## 2. 관련 연구

DARPA의 IA&S(Information Assurance and Survivability) 프로젝트는 정보전 대응에 필요한 정보 보증 및 생존 기술 개발을 위해 전략적 침입평가, 침입감내 시스템, 결함 허용 네트워크, 동적협동, 정보 보증, 정보 보증 과학 및 공학 도구, 자율적 정보 보증, 그리고 사이버 지휘 통제 등 8가지 영역에 걸쳐 기술을 개발하고 있다[2].

이 장에서는 DARPA IA&S의 침입감내 시스템 프로그램과 DARPA에서 최근 새롭게 시작된 OASIS(Organically Assured and Survivable Information Systems Program) 프로젝트에 대해 알아본다.

### 2.1 DARPA IA&S 침입감내 시스템 프로그램

과거에는 정보시스템의 보안을 위해 침입자와 데이타의 격리를 통한 기밀성과 무결성을 강조하였다. 이를 위해 중요한 정보를 암호화하는 방법과 인증된 사용자에게만 권한을 부여하는 접근 제어 방법이 많이 사용되었다.

그러나 이러한 방법은 성능과 기능성이 저하되고 암호화와 인증을 위한 비용이 추가되는 점과, 특별한 하드웨어, 소프트웨어 설계가 필요하다는 단점이 있다. 대부분의 소프트웨어 시스템은 COTS(Commercial Off-The-Shelf) 컴포넌트로 구성되어 있고 이러한 COTS 컴포넌트는 기존의 기밀성, 접근제어 방식을 적용하기 적절하지 않은 경우가 많다.

이러한 문제를 해결하기 위해 공격이 성공하더라도 중요한 서비스가 지속적으로 제공되는 것을 보증할 수 있는 가용성과 무결성을 강조하는 침입감내 시스템이 제안되었다.

DARPA에서 추진하고 있는 침입감내 시스템 프로그램의 목적은 침입에 저항성을 가지고 결함을 허용하는 시스템의 개념, 설계, 개발, 검증 아키텍처, 방법론에 대한 기술을 개발하는 것이다. 침입감내 메커니즘은 시스템 상위 계층에서 성공한 공격을 탐지하여 중요한 어플리케이션이 올바르게 수행하도록 의심스러운 코드 실행을 방지하거나 악의적인 의도로 시스템 자원에 접근하는 것을 방지하는 것이다[3].

### 2.2 DARPA의 OASIS 프로젝트

OASIS는 3세대 보안 매커니즘으로 1세대와 2세대 메커니즘을 보완하고 공격에 대한 저항을 다중 계층 형태로 제공한다[1].

저항 계층의 첫부분은 실시간 실행 모니터 부분으로 보안 정책을 위반하는 코드의 실행을 방지한다. 그러나 만약 그런 코드가 실행된 경우 다음 저항 계층이 에러의 탐지와 전파를 통해 피해를 방지한다. 추가적인 계층으로 에러 상쇄와 피해 복구, 자원의 재설정 계층이 있다.

OASIS는 특정 공격에 대한 감내 기술을 만들고 검증하여 OASIS에서 개발된 여러 감내 기술을 통합하는 과정을 거친다. 추가적으로 DARPA에서 추진되었고 추진 중에 있는 IS(Information Survivability)와 IA&S에서 개발된 결과의 통합에 대한 연구도 수행한다.

DARPA의 OASIS 프로젝트와 관계된 프로젝트로 OASIS DEM/VAL(Integration, DEMonstration and VALidation) 프로젝트[4]는 1세대, 2세대 보안 기술과 OASIS 기술의 통합에 대한 연구를 수행하고 있다. OASIS DEM/VAL 프로젝트는 2002년 8월에 시작하였으며, 2년안에 현장 시험이 가능한 프로토타입 개발을 목표로 하고 있다.

## 3. 침입감내 시스템의 분류

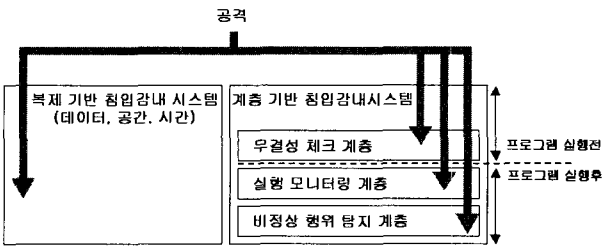
이 장에서는 침입으로부터 보호해야할 대상인 프로그램과 데이터의 관점과, 침입감내에 대한 기술적인 접근방법인 계층기반과 복제 기반 관점에서 침입감내 시스템의 분류 기준을 제시한다.

복제기반과 계층기반을 프로그램과 데이터로 다시 분류하는 이유는 프로그램과 데이터관점에서 필요로 하는 기술적 요구사항이 상이하기 때문이다. 따라서 프로그램과 데이터 관점에서 복제기반/계층기반 침입감내 시스템으로 네 가지로 분류하고 각 분류에 맞는 기술적 요구사항을 기술한다.

복제 기술은 결함허용의 메커니즘으로 많이 사용하고 있다. 그러나 단순한 복제의 증가는 기밀성에 대한 위협이 증가하므로 기존의 복제기술에 보안요소를 포함하는 방향에 대한 모색이 필요하다[5].

계층기반을 분류의 기준으로 제시하는 이유는 공격이 성공하는 단계에 따라 적절한 감내 메커니즘이 제공되어야 하기 때문이다[6]. 또한 모든 서비스에 동일한 감내 메커니즘 계층을 적용하는 것이 아니라 서비스의 중요도에 따라 감내 메커니즘 계층을 상이하게 적용하여 서비스의 성능과 침입감내 기능성을 서비스 환경에 맞게 설정할 수 있다.

(그림 1)은 계층구조와 복제에 기반한 침입감내 시스템 구조를 간략하게 분류한다.




(그림 1) 침입감내 시스템 아키텍처 분류

### 3.1 복제기반 침입감내 시스템

복제기반 침입감내 시스템은 기존의 결함허용 기술과 유사하게 분산 컴퓨팅 환경에서 침입이 발생하더라도 지속적

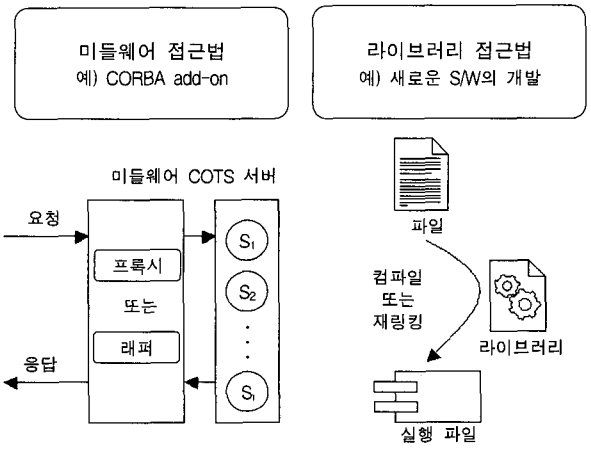
인 시스템 서비스를 제공하기 위해 가용성 확대를 목적으로 하는 시스템이다. 복제기반 침입감내 시스템은 프로그램과 데이터 관점으로 분리할 수 있다.

3.1.1 프로그램을 위한 복제기반 침입감내 시스템

결함허용 관점에서는 분산 컴퓨팅에서 부하조정을 위해 프로그램의 복제 기술을 연구하였고, 보안 관점에서는 서비스 거부 공격에 저항력있는 서비스 제공을 위해 복제 기술을 연구하였다.

프로그램을 위한 복제기반 침입감내 시스템과 유사한 형태는 클러스터링 컴퓨터이다. 클러스터는 다수의 PC 또는 워크스테이션을 연결하여 하나의 시스템으로서 함께 작동하는 컴퓨터를 말한다[7]. 클러스터링의 목적은 컴퓨팅 성능의 향상 또는 가용성의 향상으로 볼 수 있다. 이중 고가용성을 위한 클러스터 컴퓨터는 웹서버나 이메일 서버와 같이 서비스가 중단되어서는 곤란한 분야에서 사용된다. 그러나 클러스터를 이용한 복제의 증가는 취약성이 증가하여 보안을 위협하는 요소가 된다[5]. 그러므로 현재의 클러스터링 기술로 가용성을 향상하는 방법은 공격에 대한 저항력이 약한 문제점을 가지고 있다.

복제 기술은 분산 컴퓨팅 환경에서의 미들웨어에서 사용될 수 있다. 프로그램 복제를 위한 방법은 기존의 COTS 서버를 사용하기 위해 미들웨어에 래퍼 계층을 새로 포함하는 방법이 있고, COTS 서버가 아닌 경우로 새로운 소프트웨어를 개발하는 경우 미들웨어에 맞게 개발을 할 수도 있지만 미들웨어를 사용하지 않고 라이브러리를 이용하여 개발을 수행할 수 있다. (그림 2)는 프로그램을 위한 복제기반 침입감내 시스템에서 분산 컴퓨팅 환경의 미들웨어에 대한 레퍼 접근법과 라이브러리 접근법에 대한 구성이다.




(그림 2) 복제기반 침입감내 시스템 - 프로그램

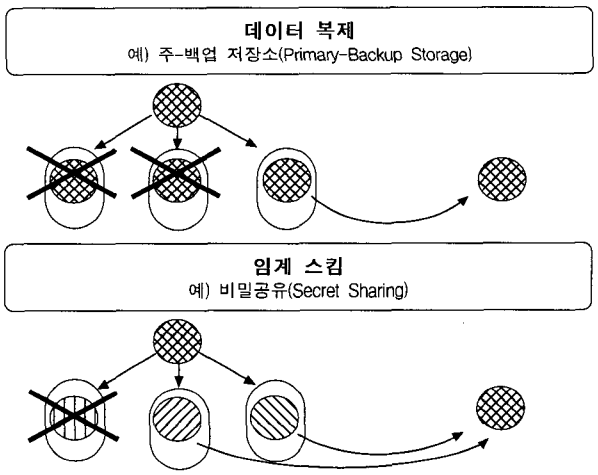
3.1.2 데이터를 위한 복제기반 침입감내 시스템

데이터의 복제는 현재의 데이터베이스나 파일 시스템에서 널리 사용되고 있다. 데이터를 위한 복제기반 침입감내 시스템에서는 기존의 안전한 스토리지 기술에서 사용되는 단순한 주-백업 저장서버의 경우에서 보다 향상된 보안성을 가지기 위해 저장서버 중 하나를 공격자가 침입 하더라도 데이터를 알아내지 못하게 하는 임계 개념(threshold scheme)이 필요하다[5]. 이런 임계 개념은 데이터를 위한 복제 기반 프레임워크 뿐만 아니라 프로그램을 위한 복제 기반 프레임워크에서도 적용하여 공격자가 하나의 서버에 침입을 성공하더라도 적절한 서비스를 제공받지 못하게 보안성을 향상시킬 수 있다.

그러므로 데이터 복제의 관점에서는 주-백업 저장서버와 같이 단순 데이터 복제와 기밀성을 강조한 임계 개념을 포함한 데이터 복제로 분류할 수 있다.

(그림 3)에 데이터에 대한 복제에 기반한 침입감내 프레임워크의 구성을 표현하였다. 그림 윗쪽은 단순 데이터 복제 기법을, 그림 아랫쪽은 향상된 보안을 지원하는 임계 개념을 포함하는 비밀공유(secret sharing) 기법을 보여준다. 비밀공유의 개념은 중요한 정보를 여러 조각으로 나누어 여러 사람이 관리하여 비밀 정보를 복원하기 위해서는 다수의 정보조각이 모여지지 않으면 비밀정보를 복원할 수 없게 하는 개념이다[5].

임계 개념은 단순 주-백업 저장시스템에 비해 가용성을 저해하지 않으면서 기밀성을 향상시키는 장점이 있고 하나의 저장소에 대해 저장공간이 늘어난다는 점과 자료를 읽거나 쓰는 지연시간이 증가하는 단점이 있다.




(그림 3) 복제기반의 침입감내 시스템 - 데이타

3.2 계층기반 침입감내 시스템

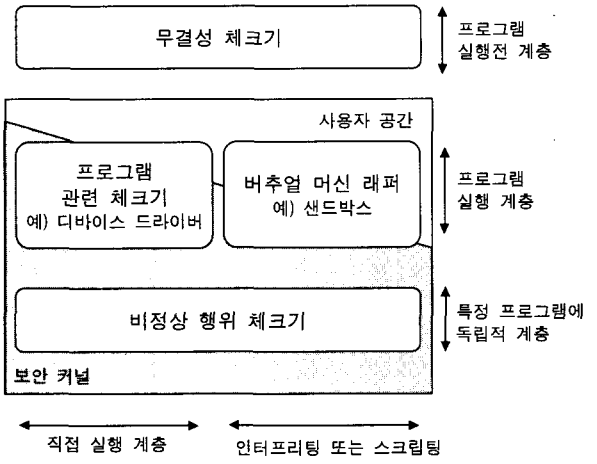
복제기반 침입감내 시스템은 분산 컴퓨팅 환경에 적용하고 계층기반 침입감내 시스템은 단일 호스트에 적용한다.

3.2.1 프로그램을 위한 계층기반 침입감내 시스템

프로그램을 위한 계층기반 침입감내 시스템에서 계층은 프로그램의 실행 전 계층, 프로그램의 실행 계층, 마지막으로 프로그램의 실행과 독립적인 계층의 세부분으로 나눌 수 있다. 첫 번째 계층은 프로그램의 실행이 이루어지기 전

에 실행 파일의 무결성을 체크하는 실행 전처리 부분이고 두 번째 계층은 실제 프로그램이 동작 중인 순간에 적합한 동작을 하는지와 악의적인 행위에 대한 중지 및 대응을 담당하는 프로그램 실행 모니터링 계층이고, 세 번째 계층은 특정 프로그램 실행과 관계없이 현재 시스템 상태를 보고 자원의 재할당과 자원에 대한 접근을 제한하는 비정상 행위 탐지 계층이다.

(그림 4)는 프로그램을 위한 계층기반 침입감내 시스템을 보인 것이다.




(그림 4) 계층기반의 침입감내시스템 - 프로그램

프로그램을 위한 계층기반 침입감내 시스템은 세로축으로 실행 전처리, 실행 중 처리, 프로그램 실행과 관계없는 커널 내부에서의 처리로 나눌수 있다. 이때 특정 프로그램 실행에 관계된 처리는 (그림 4)에서 가로축으로 실행 파일이 직접 실행되는 경우와 Java 바이트 코드와 자바 가상기계, Tcl, Perl과 같이 인터프리터에 의해 실행되는 경우, 또는 JavaScript, VBScript와 같이 스크립트로 동작하는 경우로 나눌 수 있다.

(1) 무결성 체크 계층

계층기반 침입감내 시스템에서의 첫 번째 계층인 무결성 체크 계층은 코드와 데이터가 실행전에 손상되었는지 손상되지 않았는지 구별하는 계층이다.

(2) 실행 모니터링 계층

실행 모니터링 계층은 악의적인 코드가 시스템에 피해를 주지 못하도록 하는 계층이다. 이 계층에서 고려해야 할 사항은 실행 모니터가 COTS 소프트웨어를 실행하는 경우 COTS 소프트웨어의 변경을 최소화하고, 성능저하를 최소화하고 다양한 보안 정책의 적용이 가능해야 한다는 점이다.

또한 직접 실행되는 프로그램이 아닌 경우로서 인터프리터, 스크립트로 실행되는 경우는 샌드박스를 제공해주는 것이다. 샌드박스 개념은 자바 애플릿에서의 보안 개념으로 악의적인 기능은 수행할 수 없는 영역을 제공해 주는 것이다[8].

(3) 비정상 행위 탐지 계층

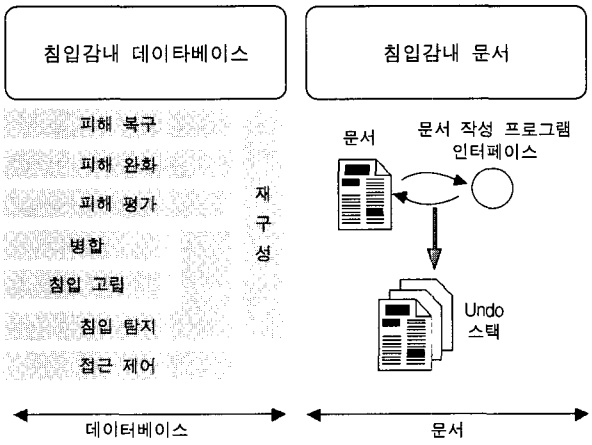
프로그램 실행에 독립적으로 운영체제의 상태에 따라 자원의 재설정을 위해서는 운영체제 내부에 해당 기능이 포함되어야 한다.

비정상 행위탐지 계층은 앞의 두 계층에서 공격을 막지 못한 경우에 동작하는 감내 메커니즘으로 운영체제내에서 비정상 행위가 탐지되면 특정 프로세스 수행과 독립적으로 공격 상황에 대한 대응을 수행한다. 예로 서비스 거부 공격의 대응은 호스트내에 특정 프로세스 수행과 독립적으로 이루어져야 한다. 이 계층에서 고려되어야 하는 사항은 다양한 비정상행위를 탐지해야 하고, 비정상행위로 오탐하는 비율이 낮아야한다.

3.2.2 데이터를 위한 계층기반 침입감내 시스템

데이터를 위한 계층기반 침입감내 시스템에 대한 연구는 프로그램을 위한 계층기반 침입감내 시스템에 비해 활발히 이루어지지 않고 있다. 왜냐하면 1, 2세대 보안 메커니즘(사용자 인증, 접근 제어 등)으로 충분히 데이터에 대한 기밀성을 보장해 주기 때문이다.

그러나 데이터에 대한 가용성의 강화는 복제기반 접근법으로 가능하지만 무결성에 대한 강화를 위해서는 계층기반 접근법이 필수적이다[9]. (그림 5)에 데이터를 위한 계층기반 침입감내 시스템이 표현되어 있다.




(그림 5) 계층기반의 침입감내 시스템 - 데이터

데이터베이스에 대한 계층기반 침입감내는 트랜잭션의 특성상 악의적인 트랜잭션이 발생한 경우 피해가 확산된다는 문제를 해결하기 위한 것이다. 그러므로 데이터베이스에도 트랜잭션 레벨에서 계층기반 침입감내 시스템이 필요하다. 현재 COTS 데이터베이스에 래퍼를 제공하여 트랜잭션 레벨의 보안을 제공하는 연구가 진행되고 있다[10].

이외의 데이터를 위한 계층기반 침입감내 시스템의 예는 문서에 대한 계층기반 접근법을 적용한 연구도 이루어지고 있다[9]. 문서에 대한 계층기반 접근은 문서작성에 대한 감리기능과 문서의 무결성을 강조하는데 목적이 있다. 문서에

대한 무결성을 강조하기 위해 문서 작성 프로그램의 인터페이스를 가로채어 문서 변경의 감리 기록을 남긴다. 결국 프로그램의 인터페이스를 가로채는 기능은 프로그램을 위한 계층기반 침입감내 특성과 동일하나 문서의 무결성을 강조하기 위해 본 분류에서는 데이터를 위한 계층기반 침입감내 시스템으로 분류하였다. 문서 작성에 대한 감리 정보에는 문서의 확대 축소와 같이 무결성에 변경을 주지 않는 기능은 포함되지 말아야 하고 문서 열기 인터페이스에 대해서는 문서를 열기 전에 문서의 무결성을 체크하여 해당 문서가 다른 곳에서 변경되지 않았음을 확인할 수 있는 기능이 필요하다.

## 4. 침입감내 시스템 사례연구

본 장에서는 3장에서 제시한 복제기반/계층기반, 데이터/프로그램에 대한 침입감내 시스템에 대한 사례연구를 수행한다.

### 4.1 프로그램을 위한 복제기반 사례연구

미들웨어에 대한 래퍼를 제공하는 접근법은 QuO(Quality Object)[11]와 ICM(Intelligent Compensating Middleware)[12]이 있다.

QuO는 어플리케이션의 코드에 직접 방어메커니즘을 포함하면 재사용성이 나빠지기 때문에 미들웨어 차원에서 이러한 매커니즘을 제공한다. QuO는 CORBA에 기반하여 분산 객체 컴퓨팅 환경에 적용할 수 있도록 CORBA의 상위집합으로 클라이언트와 스텁(stub)사이의 IDL(Interface Definition Language) 위층에 래핑되어 들어가게 된다. 클라이언트 측에서 래핑된 QuO는 서버와의 연결을 평가하여 성능, 통계, 검증 기능을 수행한다. 서비스를 제공하는 서버 측에서의 QuO는 클라이언트의 요구가 IIOP(Internet Inter Orb Protocol)를 가로채어 그룹 통신과 복제를 수행할 수 있는 게이트웨이 컴포넌트를 지원한다.

ICM은 CORBA를 이용하여 XML 기반 메시지 전송에 대한 미들웨어를 제시하였다. ICM에서는 위협의 정도에 따라 데이터에서만 임계 개념을 적용한 것이 아니라 실행 코드에서도 임계 개념을 적용하였다.

미들웨어에 대한 래퍼를 제공하는 방법 외에 프로그램 복제를 지원하는 라이브러리 방법으로는 Antigone[13] API가 있다. 이 API에는 동적협동에 포함되는 인증, 접근제어, 키 관리, 구성원 관리, 데이터 보안 기능을 제공하고 결함 탐지와 회복 기능을 포함한다. Antigone에서는 그룹 인터페이스가 어플리케이션에 위의 기능을 제공한다. 그룹 인터페이스 내부에는 보안 정책 처리를 담당하는 엔진이 존재하고 그 하부에 통신을 담당하는 브로드캐스트 통신 계층이 있다. 이와 같이 프로그램의 복제를 제공할 수 있는 라이브러리를 구축할 때에는 이러한 동적협동 기술뿐만 아니라 복제에 해당하는 기능도 필수적으로 포함되어야 한다.

### 4.2 데이터를 위한 복제기반 사례연구

PASIS[5]에서 일반적인 임계 개념에 대한 분석과 이를 적용한 PASIS 저장시스템과 단순한 주-백업 저장시스템에 대한 비교를 수행하였다. 일반적인 임계 스킴은 p-m-n 임계 개념으로 표현될 수 있는데 이 의미는 하나의 데이터에 대해 n개의 공유로 나누어지고 m개의 공유가 모여야 원래 데이터로 재구성할 수 있고, m보다 작거나 같은 p-1개가 모였을 때는 원래의 데이터를 드러내지 않는 경우를 의미한다.

<표 1>에 몇가지 경우의 임계 개념이 표현되어 있다.

〈표 1〉 임계 개념의 몇가지 경우

| 파라메터 | 설 명 |
|---|---|
| 1-1-n | 복제(Replication) |
| 1-m-n | 정보 분산(Information Dispersal) |
| m-m-n | 비밀 공유(Secret Sharing) |

N-way 복제의 경우 1-1-N의 임계 개념을 가지게 된다. 하나의 복제는 인코드된 데이터에 대한 정보를 드러낼 수 있고(p = 1), 하나의 복제만 가지고도 원래 데이터로 재구성할 수 있고(m = 1), 원래의 데이터로 재구성하고자 할 때 N개의 복제 중에 선택하기 때문이다(n = N).

PASIS에서 임계 개념은 암호학적 기법과 결합하여 이용할 수 있다. 매우 큰 파일을 복제하는 경우 단축 비밀 공유(short secret sharing)를 사용하여 원래 파일을 임의의 키로 인코딩하고, 이 암호화 키를 비밀 공유 방법으로 보관하고 암호화된 파일은 정보분산을 이용하여 관리한다.

### 4.3 프로그램을 위한 계층기반 사례연구

프로그램을 위한 계층기반 침입감내 시스템의 첫 번째 계층은 실행전 실행 파일의 무결성을 확인하는 무결성 체크 부분이다. 이와 관련된 대표적인 연구로는Tripwire[14]가 있다. 그러나 Tripwire의 운용시 실행파일의 무결성 체크 값을 단일 호스트에 보관하는 경우 침입자가 실행파일의 무결성 체크 값을 변조할 수 있는 위험이 있다. 또한 프로그램 실행때 마다 자동적인 무결성 체크를 위해서는 프로세스 실행 시스템 콜을 가로챌 수 있는 기술과의 통합이 필요하다.

Tripwire에 비해 향상된 방법으로 Emu(Execution management utility)[15] 연구가 있다. Emu는 Windows기반에서의 실행 파일의 무결성을 검사하여 실행을 할지 말지를 결정하는 것으로서 클라이언트 서버구조로 구성되고, 서버에서 관리하는 ECL(Execution Control List)을 이용하여 프로그램의 실행에 대한 결정을 내리게 된다. 프로세스를 시작하는 시스템 콜을 가로채는 디바이스 드라이버와 서버와 통신을 통해 수행을 결정하는 서비스 프로그램으로 구성된다. 파일명으로 그 파일을 실행할지 안할지를 결정하는 경우에는 사용자가 실행 파일명을 실행 가능한 이름으로 변경하면 모든 실행 파일을 실행할 수 있는 문제점이 존재하므로 ECL에는 실행파일명과 그 실행파일의 무결성을 보장할 수

있는 해쉬 값을 같이 저장하게 된다.

위에서 언급한 Tripwire와 Emu의 경우는 단순히 실행파일의 무결성을 체크하여 프로그램을 실행할 것인가 말 것인가를 결정하는 하는 방법을 취하고 있다. KLW(Kernel Loadable Wrapper)[16]는 리눅스 기반의 보안과 신뢰성을 제공하기 위한 래퍼를 제공한다. KLW는 네트스케이프 웹 브라우저 래퍼, 아파치 웹 서버 레퍼, 어플리케이션의 파일 변화에 자동적으로 파일을 복제하는 복제 래퍼 등 세 가지 레퍼를 제공한다. 세 가지 레퍼중 복제 래퍼 부분은 계층기반 침입감내 시스템이 아닌 복제기반 침입감내 시스템 부분이다. KLW의 예와 같이 효과적인 침입감내 특성을 가지려면 복제기반과 계층기반의 적절한 통합이 필수적이다.

스크립팅에 샌드박싱을 제공하는 연구[17]에서는 MS의 인터넷 익스플로러에서 JScript와 VBScript을 위한 액티브 스트립트 엔진에 대한 샌드박싱을 제공해주는 연구가 진행 중이다. MS의 스크립트 엔진은 COM 구조로 이루어져 있다. 이 연구에서는 스크립트 인터프리터 엔진과 호스트 사이트 오브젝트 사이에 정책 강화 COM 객체를 삽입한다. 정책 강화 COM 객체에 다양한 보안 정책을 적용하여 스크립팅에 대한 샌드박싱 기능을 제공하는 방법이다.

### 4.4 데이터를 위한 계층기반 사례연구

데이터베이스에서 침입에 대응하기 위한 침입감내 메커니즘을 포함하는 연구로는 ITDB(Intrusion Tolerant Database)[10]가 있다.

기존의 전통적인 데이터베이스는 보안을 위해 단순한 접근 제어를 이용하고 있으나 ITDB에서는 트랜잭션 레벨에서 침입감내 특성을 지원하여 악의적으로 승인된 트랜잭션에 의한 피해를 감내한다. 악의적인 트랜잭션을 막기 위해 침입탐지기의 성능이 중요하다. 침입탐지기에 의해 침입이 식별되면 피해를 평가하는 시간동안 피해억제기에 의해 피해가 확산되는 것을 방지한다. 침입 탐지의 정확성을 높이기 위해서는 긴 탐지 지연시간이 필요하다. 그러나 긴 탐지 지연 시간동안 악의적인 트랜잭션은 피해를 전파하는 문제가 있다. ITDB에서는 피해억제 방법이외에도 독립관리기를 통해 의심스러운 트랜잭션의 실행을 고립하여 추후에 피해가 발생한 경우 피해를 복구할 때 잃어버리는 데이터가 없게 만든다. 어플리케이션에게 침입감내 특성을 투명하게 제공하기 위해 어플리케이션은 정책강화 관리기를 통해서 데이터베이스 질의를 보내고 정책강화 관리기는 사용자 트랜잭션에 대한 프록시 역할을 수행하고 보안 정책을 적용한다.

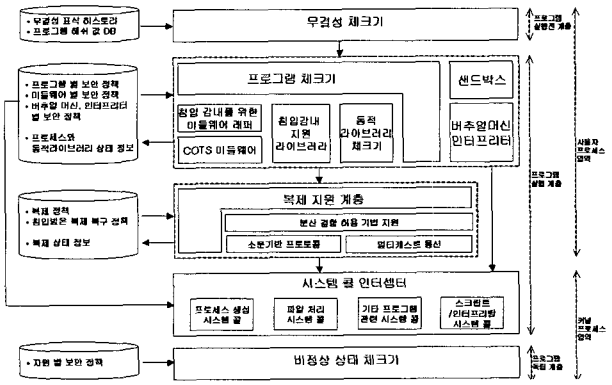
일반적으로 문서보호를 위해서 문서에 단순히 비밀번호를 부여하여 문서에 대한 접근제어를 제공한다. 그러나 DIM(Document Integrity Manager)[9]의 경우 MS Word에 해당하는 COM객체의 API를 가로채어 문서에 대한 어플리케이션 차원의 기록을 남기는 연구를 수행중이다. 이러한 기록을 바탕으로 파괴된 문서에 대한 복구를 도울 수 있다. DIM는 MS Word의 메뉴에 대한 명령, 키보드 쇼트컷, 텍스트의 드래깅, 문서의 저장과 닫기 등에 대한 이벤트에 대한 추적을 남길 수 있다. 이렇게 많은 기록을 남기는 Word의 복잡성 때문에 DIM는 문서의 상태 기록의 효율성을 위해 Word의 Undo 스택을 저장한다.

## 5. FORLIFE 프레임워크

이 장에서는 운영체제, 사용자 프로그램 및 데이터에 3세대 방어개념인 침입감내을 제공하기 위한 전사적인 틀인 FORLIFE(Framework Of Redundant Layer-based Intrusion and Fault Endurance)를 제안한다.

### 5.1 FORLIFE 프레임워크의 구성

복제기반 침입감내 및 계층기반 침입감내는 서로 상호보완적 성격을 가지기 때문에 FORLIFE 프레임워크는 복제기반 침입감내 시스템과 계층기반 침입감내 시스템의 기능을 모두 가지도록 구성한다. 예를 들어 복제 기반 침입감내 시스템 프레임워크는 분산 컴퓨팅 환경에서 복제를 통해 가용성을 향상하고 각각의 노드는 계층기반 침입감내 시스템이 되어 보다 강화된 침입감내 특성을 가지게 할 수 있다.




(그림 6) FORLIFE 프레임워크

(그림 6)에 FORLIFE 프레임워크를 표현하였다. 침입감내 시스템 프레임워크는 기본적으로 계층기반과 복제 기반이 계층구조로 통합되어 있다. 이러한 계층구조의 장점은 도메인에 맞는 침입감내 특성을 제공할 수 있도록 설정을 조절할 수 있다는 점이다. 예를 들어 사용자가 침입감내 시스템에서 복제를 사용하지 않는 경우 복제지원 계층을 생략하면 된다.

FORLIFE는 무결성 체크기, 프로그램 체크기, 샌드박스, 시스템 콜 인터셉터, 비정상 상태 체크기는 계층기반 침입감내 기능을 제공하고 복제 지원 계층은 복제기반 침입감내 시스템의 기능을 제공한다.

### 5.2 FORLIFE 프레임워크 구성요소의 기능

본 절에서는 FORLIFE의 구성요소에서 제공하는 기능을

〈표 2〉 침입감내 시스템 분류기준과 FORLIFE 기능 연관성

| 분 류 기 준 | 사 례 연 구 | FORLIFE 프레임워크 기능 |
|---|---|---|
| 복제기반 침입감내 시스템 - 프로그램 | QuO, ICM, Antigone | ● 복제지원 계층<br>• 통신 방법(멀티캐스트, 소문 프로토콜) 기능<br>• 임계 개념 적용 기능 |
| 복제기반 침입감내 시스템 - 데이터 | PASIS | |
| 계층기반 침입감내 시스템 - 프로그램 | Tripware, Emu, ECL, KLW, Interception Wrapping and Anaysis Framework for Win32 Scripts, Characterzing Intrusion Tolerant Systems Using a State Transition Model | ● 무결성 체크기<br>• 단일호스트 무결성 체크 또는 C/S 무결성 체크 기능<br>● 프로그램 체크기<br>• 실행파일 체크 기능<br>• 인터프리터를 위한 샌드박싱 기능<br>● 시스템 콜 인터셉터<br>• 디바이스 드라이버를 이용한 인터셉트<br>• 라이브러리를 이용한 인터셉트<br>● 비정상 상태 체크 |
| 계층기반 침입감내 시스템 - 데이터 | ITDB, DIS | |

설명한다. <표 2>에 3장 침입감내 분류, 4장 사례연구, 5장의 FORLIFE 프레임워크의 연관성이 표현되어 있다.

5.2.1 무결성 체크기

이 부분은 계층기반 침입감내 시스템의 세부 분류에서 첫 번째 계층에 해당하고 실행파일이 침입으로 인해 변경이 된 경우를 발견하기 위한 무결성 체크 부분이다.

무결성 체크기는 실행 파일의 무결성을 체크하여 무결성이 변경된 경우 사용자에게 변경되었다는 것을 공지하여 실행 여부를 사용자가 결정하는 정책을 적용할 수도 있고 무결성이 변경되면 자동으로 실행을 하지 않는 정책을 적용할 수도 있다.

또한 무결성을 보장하기 위한 해쉬값을 로컬에서 보관하는 정책과 중앙집중적인 서버에 해쉬값을 저장하는 정책을 선택할 수 있다. 중앙집중적인 서버에 보관을 하면 각각의 호스트에서 요구하는 실행파일의 실행여부를 중앙 서버에서 알려주게 되고 무결성 체크을 위한 관리가 용이해진다[15]. 이 경우 각각의 호스트에 존재하는 실행파일이 동일한 기능을 수행하는 것을 보장하기 위해서 실행파일 이름, 버전 정보, 벤더 정보도 함께 저장되어야 한다.

무결성 체크기가 자동적으로 실행되기 위해서는 시스템 콜 인터셉터 계층에서 유닉스의 경우 execle 패밀리, fork, Win32 API의 경우 CreateProcess와 같은 프로세스를 생성하거나 프로세스 문맥을 변경하는 시스템 콜을 자동으로 가로채어야 한다. 왜냐하면 실행파일에 대한 무결성 체크는 실행하기 바로 전에 자동적으로 실행하는 것이 가장 정확하기 때문이다.

5.2.2 프로그램 체크기

이 부분은 계층기반 침입감내 시스템의 세부 분류에서 두 번째 분류에 해당하고, 프로세스 문맥이 제시된 정책에 맞게 수행하는지 확인하는 프로그램 체크기, 동적 라이브러리 체크기의 기능과 미들웨어와 프로세스가 침입을 받은 것이 탐지된 경우 서비스의 안전한 중단, 복구 기능을 제공하는 미들웨어 래퍼와 침입감내 지원 라이브러리 기능, 인터프리터와 스크립팅을 위한 샌드박스 기능으로 구성된다.

이 계층에서 선택가능한 설정은 두가지 이다. 첫 번째는 각각의 프로그램에 관련된 구체적인 보안 정책이 제시되지 않은 경우로서 이 경우는 무결성 체크기에서 실행파일이 호출하는 시스템 콜에 대한 정적 정보와 DLL에서 호출하는 시스템 콜 정보를 정적으로 체크하여 여기에 해당하지 않는 시스템 콜이 호출된 경우 해당 시스템 콜을 수행하지 않는 기능이다.

두 번째는 각각의 프로그램에 보안 정책이 제공되는 경우로서 이 경우는 첫 번째 설정에서 무결성 체크기에서 알아내는 정보는 필요하지 않고 프로세스에 시스템 콜 수준의 보안 정책이 명시적으로 제시된 경우 보안정책을 위배하는 시스템 콜을 호출하는 경우 해당 시스템 콜을 수행하지 않는 방법이다. 이 경우 프로그램의 기능을 명시적으로 알고 있어 각 기능에 대한 시스템 콜을 명세할 수 있는 경우에 유용하다.

실행파일이 아닌 인터프리터를 위한 샌드박스 기능을 제공하기 위해서 자바 가상기계에서의 직접적인 보안 정책 설정 방법과 액티브 스크립트 엔진을 위한 COM 인터페이스를 가로채는 방법과 같이 각각의 인터프리터에 맞는 샌드박스 기술도 제공되어 인터프리터를 위한 통합 관리기능이 있어야 한다.

5.2.3 복제 지원 계층

이 부분은 복제기반 침입감내 시스템에 해당하고 가용성을 강화하기 위해 적용한다. 이 계층에서는 주로 파일처리에 관계된 API를 인터셉트하여 일반 파일처리에 관계된 함수가 복제를 지원하도록 확장하는 것이다. 필요한 기능으로는 분산 결함허용 기능, 분산 통신 기능, 임계 개념을 지원하는 복제 기능으로 나눌수 있다.

이 부분에서 선택가능한 설정은 기존의 분산 컴퓨팅 환경에서의 다양한 결함 허용 기법에 대한 설정과 분산 환경을 지원하기 위한 통신 부분으로 멀티캐스트 통신을 이용할 것인지 소문 프로토콜(gossip protocol)을 사용할 것인지에 대한 부분이다. 소문 프로토콜은 한 그룹의 멤버들이 소문을 이용하여 서로에게 정보를 전달하여 모든 멤버가 동일한 정보를 공유할 수 있도록 하는 프로토콜이다[18]. 멀티캐스트는 확장성에 그리 좋은 성능을 보이지 않으므로 대규모의 분산 환경에서는 소문 프로토콜 방법이 더욱 바람직하다. 또한 가용성을 강조하기 위해 복제 기법을 사용하더라도 복제되는 데이터가 중요하다면 임계 개념을 적용할 수도 있다. 예를 들어 중요한 데이터인 경우 비밀 공유를 적용하고 큰

데이터인 경우 단축 비밀 공유를 사용할 수 있다.

5.2.4 시스템 콜 인터셉터

이 부분은 계층기반 침입감내 시스템의 세부분류로 두 번째 계층에 해당하고 무결성 체크기을 위해 프로세스의 수행에 관계한 시스템 콜 정보를 가로채고 프로그램 체크기에 다양한 시스템 콜 정보를 전달하고, 액티브 스크립트 엔진의 인터페이스을 가로채어 샌드박스에 그 호출정보를 전달하는 기능을 수행한다. 또한 파일 시스템에 복제 개념을 사용하려는 경우에는 파일 처리에 관련된 시스템 콜을 복제지원 계층에게 알려주는 역할도 수행한다.

이외에도 프로세스의 kill, 서비스의 중지와 관련된 시스템 콜에 대한 로그, 사용자 확인 후 시스템 콜 수행과 같은 기능도 필요하다.

시스템 콜 인터셉터를 구현하기 위한 기술적 방법은 실행파일의 임포트 테이블에 대한 변경을 하는 라이브러리를 이용하는 방법과 디바이스 드라이버를 이용하여 커널 영역으로 인터셉트 기능을 포함하는 방법으로 나누어지고 이런 기술적 분류도 선택가능한 설정이다.

MS Windows의 경우 라이브러리 형태로 시스템 콜을 인터셉트하는 Detours 라이브러리[19]를 사용하는 방법이 있다. Detours는 Windows 실행파일인 PE 파일내에 임포트 테이블을 변경하여 실행파일이 실행될 때 Win32 API을 인터셉트하는 구조이다. 리눅스와 솔라리스에서 사용되는 실행하는 파일 형식인 ELF 파일의 경우 injectso[20]을 이용하여 시스템 콜을 인터셉트 할 수 있다.

5.2.5 비정상 상태 체크기

이 부분은 계층기반 침입감내 시스템의 세부분류의 세 번째 분류에 해당하고 주기적으로 운영체제 상태를 검사하여 시간적, 공간적 비정상 상태를 체크하고 비정상 상태에 맞는 대응과 자원에 대한 보호를 제공하는 기능을 수행한다.

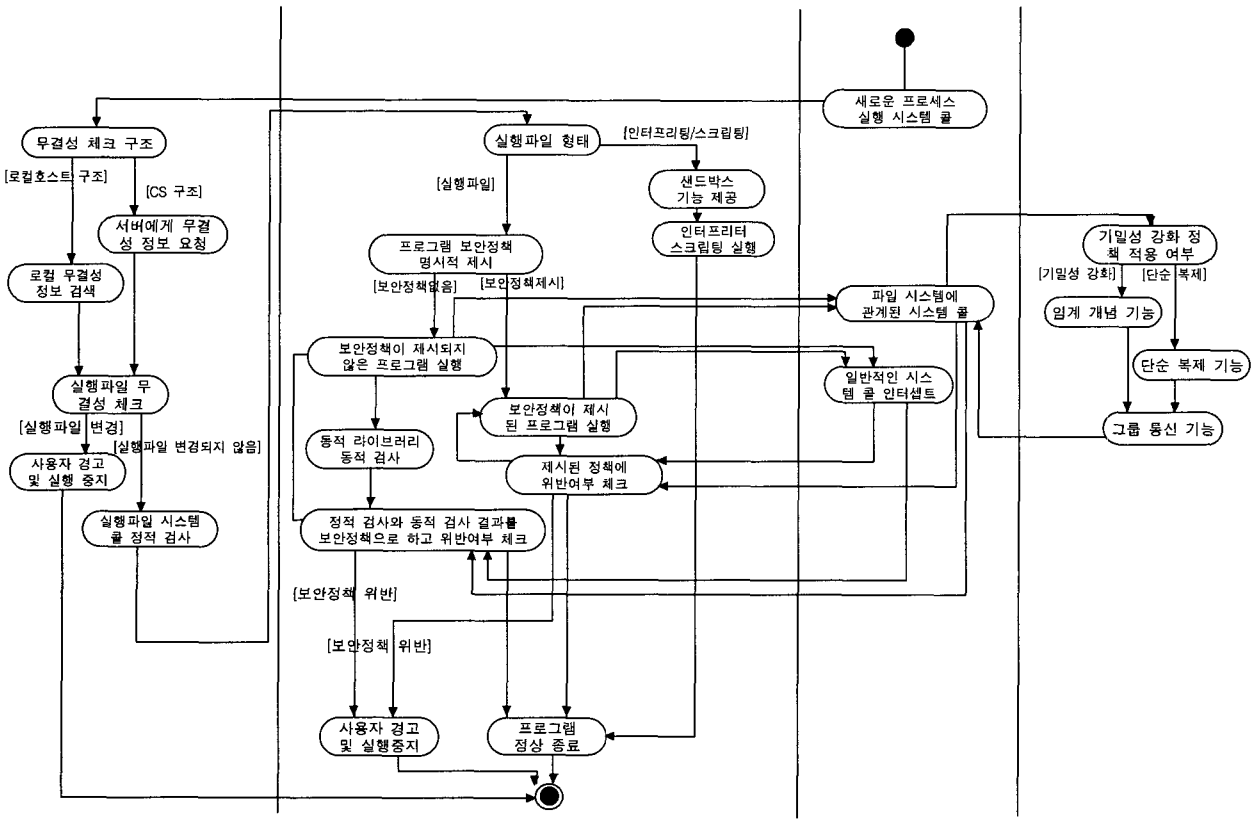
이 계층에서 선택가능한 설정은 운영체제 전체적인 설정값과 비정상으로 탐지할 임계값을 선택해야 한다.운영체제 전체적인 설정값은 TCP 연결의 불안전 대기열 수, 전체 프로세스 디스크립터의 수, 전체 프로세스들이 가진 파일 디스크립터의 수 등이다.

비정상으로 탐지할 임계값에 해당하는 부분은 하나의 프로세스가 차지할 CPU 점유률, 메모리 사용량 등이다.

5.2.6 공통 기능

이 절에서는 무결성 체크기, 프로그램 체크기, 복제지원 계층, 시스템 콜 인터셉터, 비정상 상태 체크기에 공통적으로 적용해야하는 기능에 대해 설명한다.

먼저 각 구성요소에 통합 적용할 수 있는 감사 정보를 남기는 기능이다. 이러한 감사 정보는 보안정책에 따라 사용자에게 대화상자로 실시간으로 공지를 할 수도 있고 로그 파일로 남길 수도 있다. 무결성 체크기에서는 실행파일,




(그림 7) FORLIFE 프레임워크의 액티버티 다이어그램

혹은 문서 파일의 변경 여부 기록이 감사 정보이고, 프로그램 체크기에서는 실행중인 프로세스에 대한 보안정책 적용에 대한 기록과 보안정책을 어긴 상태의 경우의 대응방법에 대한 기록이 감사정보이고, 복제지원 계층에서는 하나의 복제에 해당하는 로컬 정보와 이 로컬 정보들을 통합된 전체 복제 상황에 대한 전체 복제 정보가 감사정보이다. 예를 들어 복제 노드가 고장 혹은 침입을 받았는지에 대한 로그와 대응 방법에 대한 로그도 필요하다.

시스템 콜 인터셉트 계층에서 모든 시스템 콜마다 기록을 남기면 그 정보의 양이 방대하기 때문에 프로그램의 행위가 변경되는 시스템 콜 경우에만 감리 기록을 남기는 것이 필요하다. 마지막으로 비정상 상태 체크기에서는 주기적인 운영체제 상태에 대한 기록과 비정상 상태인 경우의 대응 방법이 감사 정보가 된다. (그림 7)에 보안 정책에 설정에 따른 FORLIFE에 적용될 수 있는 침입감내 특성을 전체적인 흐름을 액티버티 다이어그램(activity diagram)로 표현하였다.

## 6. FORLIFE 평가

현재 침입감내 시스템 기술에 대한 연구가 활발히 진행중에 있다. <표 3>에 다른 침입감내 프레임워크와 본 논문에서 제시한 프레임워크를 비교하였다. Intrusion Tolerant Software Architecture[21]는 아키텍처 기술언어를 이용하여 침입감내 메커니즘을 소프트웨어 개발 전 과정에서 고려한다는 장점을 가지고 있으나 데이터 흐름에 대한 검증을 중심으로한 침입감내 메커니즘 평가를 제외하고는 구체적인 침임감내 메커니즘을 제시해주지 못하는 단점을 가지고 있다.

SITAR[22]의 경우에는 침입탐지 시스템과 침입감내 시스템의 연동을 유기적으로 잘 표현하고 있는 점과 분산 환경에서 COTS 서버에 침입감내 메커니즘의 적용이 용이한 점을 가지고 있으나 단일 호스트에 적용을 하는 경우 무결성 체크 기능외에 다른 침입감내 메커니즘을 가지고 있지 않는 단점을 가지고 있다.

FORLIFE는 Intrusion Tolerant Software Architecture에 비해서 구체적인 침입감내 메커니즘을 제시해주는 장점과 SITAR 처럼 단일 호스트인 경우에 침입감내 메커니즘이 무결성 체크 기능만 제공해주는것 이외에도 시스템 콜 수준의 자세한 보안 정책을 적용할 수 있을 뿐만 아니라 복제와의 연동도 가능한 장점을 가지고 있다.

## 7. 결 론

침입감내 시스템은 침입방지, 침입탐지에 다음에 나타난 제3세대 보안 개념으로 악의적인 침입이 성공하더라도 시스템의 주요 기능을 지속적으로 제공하기 위하여 보안에서의 가용성과 무결성을 강조한다. 미국 DARPA의 경우 1990년대 중반까지 Information Survivability 프로젝트, 1990년대 후반까지 IA&S 프로젝트와 현재 수행중인 OASIS 프로젝트에 이르기까지 지속적으로 침입감내 시스템에 관계된 기술을 개발하여 앞으로 군 IT 장비위주로 침입감내 특성을 포함한 시스템을 설치하려고 계획하고 있다.

본 논문에서는 DARPA에서 진행중인 침입감내 시스템에 대한 분석을 통해 침입감내 시스템 전반에 필수적인 기술에 대한 체계적인 프레임워크를 제시하였다.

제안된 침입감내 프레임워크를 웹 서버에 적용한다면 웹 서버에 시스템 콜 수준의 보안정책이 반영되고 웹 서버 공격의 저항력을 높일 수 있다. 이와 같이 FORLIFE 프레임워크는 웹 서버 뿐만아니라 지속적인 서비스 제공을 위한 서버 분야에서 널리 활용할 수 있을 것이다.

침입감내 시스템의 프레임워크 단독으로는 통합적인 방어 특성을 모두 제공할 수 없으므로 향후 침입감내 시스템 뿐만 아니라 복제 기반과 계층기반에 필요한 동적협동 기술과 전략적 침입 평가기술과의 이들의 유기적인 통합에 대한 연구가 필요하다.

〈표 3〉 침입감내 프레임워크 비교

| 시스템 이름 | FORLIFE | Intrusion Tolerant Software Architecture | SITAR |
|---|---|---|---|
| 접 근 법 | • 복제기반/중복기반, 프로그램/데이타 침입감내 특성의 통합 | • 아키텍처 기술언어을 이용하여 침입감내 특성의 검증을 지원하는 아키텍처 | • 분산 서비스에서의 침입감내 메커니즘을 지원하기 위한 아키텍처<br>• 침입탐지 시스템과 침입감내 시스템의 유기적 통합 |
| 소프트웨어 개발 과정 지원 | • 침입감내 지원 미들웨어 래퍼화 침입감내 지원 라이브러리 구현에 도움 | • 요구사항 명세에서 구현까지 아키텍처 기술언어을 이용하여 정형화된 검증을 통한 소프트웨어 개발 전 과정 지원 | • 분산환경에서 서비스 제공하는 구조를 가진 도메인만 지원가능 |
| 복제기반 침입감내 시스템 기능 | • 선택가능한 설정을 이용한 단순 복제, 임계 개념을 적용한 복제와 같이 다양한 기능 제공 | • 개발자가 복제기반 메커니즘 직접 설계/구현 | • 플룩시 서버를 이용한 침입탐지 기능 제공<br>• 각 복제의 감리정보를 통합 제공 |
| 계층기반 침입감내 시스템 기능 | • 프로그램 실행전, 프로그램 실행시 프로그램 독립적인 부분에서 선택가능한 설정 제시 | • 개발자가 계층기반 메커니즘 직접 설계/구현 | • 단일 호스트내에서 침입감내 특성은 무결성 체크 기능밖에 없음 |

## 참 고 문 헌


[1] DARPA OASIS project home page, http://www.darpa.mil/ipto/research/oasis/.

[2] 박상서, 정보전 대응체계 구축 현황, WISC2000 튜토리얼 자료집, 2000.

[3] http://www.afrlsn.afrl.af.mil/IA&S_topics.htmll#ITS.

[4] http://www.darpa.mil/ipto/research/oasis/demval-goals.html.

[5] Gergory R. Ganger, et. al., Survivable Storage Systems, Proceedings of DISCEX2001, 2001.

[6] Katerina Goseva-Popstojanova, et. al., Characterizing Intrusion Tolerant Systems Using a State Transition Model, Proceedings of DISCEX2001, 2001.

[7] 유찬수, 리눅스 클러스터링, 정보과학회지, 제18권 제2호, 2000.

[8] Gary McGraw, et. al., *Securing Java*, Wiley, pp.38-48, 1999.

[9] Marcelo Tallis, et. al., Document Integrity through Mediated Interfaces, Proceedings of DISCEX2001, 2001.

[10] Peng Liu, et. al., Intrusion Tolerant Database Systems, Technical Report, Dept. of Info. Systems, Univ. of Maryland, Baltimore County, 2001.

[11] Partha P. Pal, et. al., Defense-Enabling Using Advanced Middleware : An Example, Proceedings of MILCOM2001, 2001.

[12] Amjad Umar, et. al., Intrusion Tolerant Middleware, Proceedings of DISCEX2001, 2001.

[13] Patrick McDaniel, et. al., Antigone : A Flexible Framework for Secure Group Communication, Proceedings of the 8th USENIX Security Symposium, pp.99-114, 1999.

[14] Gene H. Kim, et. al., The Design and Implementation of Tripwire : A File System Integrity Checker, Proceedings of the 2nd ACM Conference on CCS '94, pp.18-29, 1994.

[15] Matthew Schimid, et. al., Preventing the Execution of Unauthorized Win32 Applications, Proceedings of DISCEX 2001, 2001.

[16] Terrance Mitchem, et. al., Linux Kernel Loadable Wrapper, Proceedings of DISCEX2000, 2000.

[17] Tim Hollebeek, et. al., Interception, Wrapping and Analysis Framework for Win32 Scripts, Proceedings of DISCEX 2000, 2000.

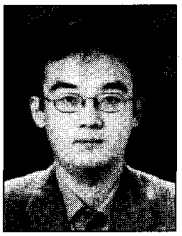
[18] Kenneth P. Birman, et. al., Spinglass : Secure and Scalable Communication Tools for Mission-Critical Computing, Proceedings of DISCEX2001, 2001.

[19] Galen Hunt, et. al., Detours : Binary Interception of Win32 Functions, Proceedings of the 3rd USENIX Windows NT Symposium, 1999.

[20] Shaun Clowes, BlackHat Briefings 2001 in Amsterdam, Holland, http://www.securereality.com.au/archives.html.

[21] Victoria Stavridou, et. al., Intrusion Tolerant Software Architectues, Proceedings of DISCEX2001, 2001.

[22] Feiyi Wang, et. al., SITAR : A Scalable Intrusion-Tolerant Architecture for Distributed Services, Proceedings of the 2001 IEEE Workshop on Information Assurance and Security, 2001.


### 김 기 한

e-mail : ghkim1@etri.re.kr

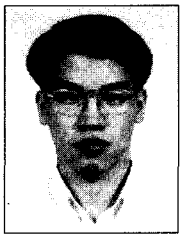
1997년 중앙대학교 컴퓨터공학과(학사)

1999년 중앙대학교 컴퓨터공학과(석사)

1999년~현재 중앙대학교 컴퓨터공학과 박사과정

2001년~현재 ETRI 부설 국가보안기술연구소 연구원

관심분야 : 정보보증, 시스템 프로그래밍, S/W 개발 방법론


### 최 명 렬

e-mail : mrchoi@etri.re.kr

1991년 인하대하교 전자계산공학과(학사)

1993년 인하대학교 전자계산공학과(석사)

1993년~2000년 국방과학연구소 선인연구원

1993년~현재 인하대학교 전자공학과 박사과정

2000년~현재 ETRI 부설 국가보안기술연구소 선임연구원

관심분야 : 정보보증, 시스템 보안, 네트워크 보안, 멀티캐스트 보안



### 이 경 환

e-mail : kwlee@object.cau.ac.kr

1964년 중앙대학교 이과대학 수학과(학사)

1966년 중앙대학교 이과대학 수학과(석사)

1980년 중앙대학교 대학원 수학과(박사)

1992년~현재 ISO/SC7/WG10 SPICE 한국 위원장

1992년~현재 한국표준소프트웨어공학(SC7) 위원

1999년~2000년 한국정보과학회 회장

1971년~현재 중앙대학교 컴퓨터공학과 교수

관심분야 : CBD Architecture, SPI(Defect Analysis), MBASE/CeBASE