침해감내 프락시 서버 프레임워크에 대한 연구*

박찬호°[1] 권영찬[1] 윤호상[2] 장희진[2] 이성기[2]
[1](주)안철수연구소
{chpark, yckwon}@ahnlab.com
[2]국방과학연구소
{yunhs, janghj, leesk}@add.re.kr

Design of Intrusion Tolerance Proxy Server Framework

Chan-ho Park°[1] Young-chan Kwon[1] Ho-sang Yun[2] Hee-jin Jang[2] Seongkee Lee [2]
[1]AhnLab, Inc.
[2]Agency for Defense Development

요 약

침해감내는 침입을 악의적인 의도를 가지고 Fault를 유발시키는 행위로 보고 이러한 공격이 성공한 경우에도 적절한 수준의 가용성, 무결성, 기밀성을 유지하면서 시스템의 중요 서비스를 일정한 시간 동안 지속적으로 제공하여 생존 성을 보장하는 것을 목표로 하고 있는 연구 분야이다. 본 논문에서는 기존의 네트워크 정보 시스템에 대한 재 구성 및 수정을 최소화하면서 주요 서비스에 대한 생존 성을 강화 시킬 수 있는 침해감내 프락시 서버 프레임워크를 제시하고 있다.

1. 서론

2002년에 발표된 자료[1]에서 미국 국방부의 연구개발 기관인 DARPA [1] (Defense Advanced Research Projects Agency)는 보안기술의 진화 과정을 3세대로 나누어 설명하고 있는데 1세대 보안기술은 암호화 기술을 중심으로 정보에 대한 접근제어와 변조를 방지하는 분야에 초점을 맞추었고 2세대 보안기술은 현재 많이 사용되고 있는 방화벽 또는 침입탐지 시스템 등을 통해서 침입을 탐지하고 공격에 의한 손상을 제한하는 방향으로 진행 되었다고 얘기하고 있다. DARPA는 3세대 보안 기술로 사이버 공격을 감내하고 공격 속에서도 중요 기능을 운용하고 유지하기 위한 기술을 제시하고 있는데 이후 이러한 3세대 보안 기술을 통해 공격이 진행되거나 심지어 공격이 성공한 경우에도 대부분의 중대한 서비스는 가용 상태로 유지될 수 있는 보안이 강화된 네트워크 정보 시스템을 만드는 것을 목표로 많은 연구들이 진행되어 왔다.

DARPA 가 제시한 3 세대 보안기술로 분류될 수 있는 침해감내는 고장 감내 기술(Fault Tolerance)을 보안분야로 확대하여 침입을 악의적인 의도를 가지고 Fault 를 유발시키는 행위로 보고 이러한 공격이 성공한 경우에도 적절한 수준의 가용성, 무결성, 기밀성을 유지하면서 시스템의 중요 서비스를 일정한 시간 동안 지속적으로 제공하여 생존 성을 보장하는 것을 목표로 하고 있는 연구 분야이다.

침해감내 기술은 시스템, 네트워크 등 다양한 관점에서 연구되고 있는데 생존 성 보장이라는 침해감내 기술의 주요 목표의 광범위함으로 인해 연구 결과를 적용하기 위해서는 많은 경우, 기존의 네트워크 정보 시스템에 대한 수정 및 재 구성이 필요하게 된다.

본 논문에서는 기존에 구축되어 운영되고 있는 네트워크 정보 시스템에 대한 수정을 최소화하면서 침해감내 서비스를 제공하기 위한 침해감내 프락시 서버 시스템에 대한 프레임워크를 제안한다.

2. 연구동향

침해감내 분야의 연구는 대부분 정부주도 형태로 진행되어 왔으며 대표적인 연구 사례로는 유럽 정보보호 연구 프로그램인 IST [2] (Information Security Technology)에서 진행한 MAFTIA [3] (Malicious-and Accidental-Fault Tolerance for Internet Applications)와 DARPA 에서 진행한 OASIS(Organically Assured and Survivable Information Systems)를 들 수 있다.

OASIS 는 외부의 공격에 대한 피해를 최소화하여 어떠한 상황에서도 일정 수준 이상의 시스템 동작이 보장되는 정보 시스템의 개발을 목적으로 진행된 프로젝트로서 다음과 같은 목표를 가지고 있다.

- 다양한 취약요소를 극복할 수 있는 침입감내 시스템의 개발
- 악의적인 코드의 복제로 인한 피해방지
- 악의적인 이동코드의 실시간 탐지
- 오류탐지, 감내, 복구, 치료기술 개발
- 침입감내 메커니즘의 평가 및 조정 방법론 개발

MAFTIA 는 대규모 분산 시스템에서 우발적인 결함(accident faults)과 악의적인 공격(malicious attacks)을 감내하기 위하여 포괄적인 접근 방법의 조사가 유도된 연구이다. MAFTIA 는 시간의 소모성이 크고 잠재적인 에러를 유발할 수 있는 사람의 개입 없이도 공격 중에서 시스템이 이전과 같이 작동하는 것을 가능하게 하고, 결함

* 이 논문은 국방과학연구소 사이버 침입탐지 및 대응기술 사업의 지원으로 작성되었음

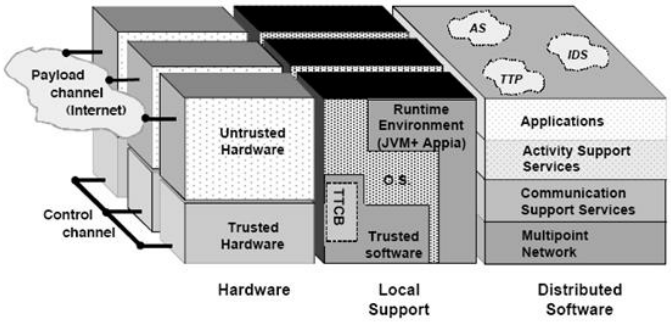
[1] http://www.darpa.mil

[2] http://www.istsecure.com

[3] http://www.maftia.org

감내, 분산 컴퓨팅, 암호 해독 법, 정형적인 증명, 컴퓨터 보안과 침입 탐지관련 커뮤니티들로부터 해당 분야의 전문지식들을 도입하여 활용한다. MAFTIA 는 보안에 대한 '감내 패러다임(tolerance paradigm)을 체계적으로 연구하여, 감내 패러다임을 기반으로 개발된 통합된 구조를 제안하고 많은 어플리케이션의 의존성(dependability)을 지원하기 위한 견고한 설계를 제공하는데 있다. 따라서 공격이 반드시 실패되도록 하는 목적보다는 공격이 발생가능하고 일정부분에 대한 몇몇 공격은 성공적일 수 있다는 가정을 기반으로 한다. 즉, 몇몇 하위 시스템에 대한 공격이 성공적이라 할지라도 전체 시스템은 안전하게 사용될 수 있도록 하기 위한 연구이며 다음과 같은 목적을 갖는다.

- 아키텍처적인 프레임워크와 개념적 모형의 정의
- 메커니즘과 프로토콜의 디자인
- 정형적인 인증과 평가

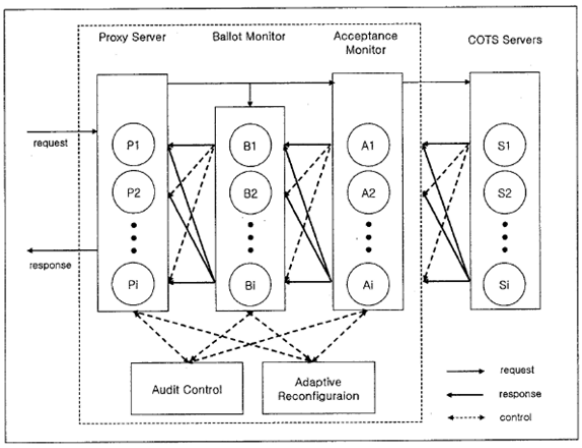


[그림 1] MAFTIA Architecture dimension

OASIS 와 MAFTIA 에서 진행된 많은 연구들은 침해감내 서비스를 제공하기 위하여 시스템 및 네트워크 인프라에 대한 재 구성 및 수정을 요구하며 이러한 요구조건들은 기존에 연구된 침해감내 기술을 현재 구축되어 운영되고 있는 네트워크 정보 시스템에 적용하는 것을 어렵게 만드는 요인이 되고 있다.

이와 같은 제약조건을 극복하기 위해 기존에 구축되어 운영되고 있는 네트워크 정보 시스템인 COTS(commercial off-the-shelf) 서버들에 대한 수정을 최소화하면서 침해감내 서비스를 제공하기 위한 연구가 진행된 것들이 있는데 대표적인 사례로 OASIS 프로젝트의 서브 프로젝트 였던 SITAR(Scalable Intrusion-Tolerance Architecture)와 DIT (Dependable Intrusion Tolerance) 가 있다.

2.1 SITAR (Scalable Intrusion-Tolerance Architecture)



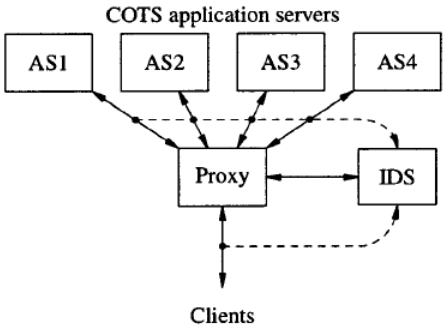
[그림 2] SITAR Architecture

SITAR 는 분산 서비스를 제공하는 고 가용성의 침입 감내 시스템을 위한 확장 성 있는 아키텍처 및 프레임워크이다. 특히 기존에 구축되어 운영되고 있는 COTS 서버를 위한 침해감내 구조를 제시하고 있으며 취약성을 보강하고 필수 어플리케이션에 대해서는 언제나 최소한의 서비스 제공이 가능하도록 하는 것을 목적으로 하고 있으며 다음과 같은 구성 요소들로 이루어진다.

- Proxy Server: 서비스 요청에 대해 침해감내 전략에 의해 규정된 서비스 정책을 적용하는데 이러한 정책은 사용자 요청을 처리할 COTS 서버의 선택과 최종 결과가 제공되는 방법을 결정하게 됨
- Acceptance Monitor: 응답에 대한 유효성을 검사하고, 선택적으로 그 결과를 Ballot Monitor 에게 전달
- Ballot Monitor: 불일치가 발생했을 때 각각의 COTS 서버의 대리인 역할을 하고, 다수결이나 Byzantine Agreement 과정을 수행함
- ARM (Adaptive Reconfiguration Module): Acceptance Monitor 와 같은 모듈로부터 침해 발생 정보를 받아 이를 평가하고, 성능 영향을 분석하며, 시스템의 새로운 구성을 생성함
- Audit Control: 주기적인 진단을 시행하여 비정상적인 행위를 파악하고 기록함

2.2 DIT (Dependable Intrusion Tolerance)

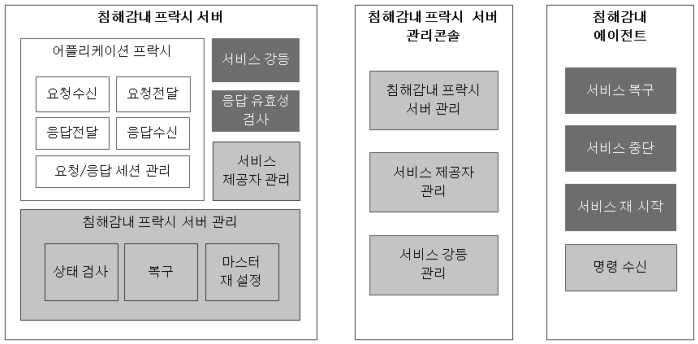
DIT 는 프로토타입 형태의 침해감내 서버 아키텍쳐를 개발하는 것을 목적으로 진행된 프로젝트이다. 침해감내 서버 아키텍쳐는 프락시 와 침입탐지 시스템으로 구성되며 이를 통해 보호받는 어플리케이션 서버들은 기존의 COTS 서버들을 그대로 사용하는 것을 가정하고 있다.



[그림 2] DIT Architecture

3. 침해감내 프락시 서버 프레임워크

침해감내 프락시 서버 프레임워크는 기존에 구축되어 운영되고 있는 COTS 서버들에 대한 수정을 최소화 하면서 침해감내 서비스를 제공한다. 다양한 서비스들에 대한 프락시들을 이중화된 침해감내 프락시 서버에 구축하고 서비스 강등 및 응답 유효성 검사 기능 등을 통하여 주요 서비스에 대한 생존 성을 유지할 수 있도록 해주는데 이때 대상 어플리케이션에 의존적인 응답 유효성 검사와 같은 기능은 공통된 인터페이스를 제공하여 대상 어플리케이션에 적합한 응답 유효성 검사 알고리즘을 구현하여 적용할 수 있는 구조를 가진다.



[그림 3] 침해감내 프락시 서버 프레임워크 구조

침해감내 프락시 서버 프레임워크는 다음과 같은 세가지 구성요소로 이루어진다.

- 침해감내 프락시 서버
- 침해감내 프락시 서버 관리콘솔
- 침해감내 에이전트

3.1 구성요소

3.1.1 침해감내 프락시 서버

서비스 요청자로부터의 요청을 대상 서비스 제공자(COTS 서버)에게 전달하는 과정에서 침해감내 서비스를 제공하는 프락시 형태의 서버 시스템으로 다음과 같은 기능을 제공한다.

- 사전에 설정된 서비스 제공자 목록에 따라 등록된 하나 이상의 서비스 제공자에게 동일한 요청을 전달.
- 어플리케이션 별로 구현된 응답 유효성 검사 API 를 통하여 서비스 이상 유무 식별
- 사전에 정의된 강등 정책에 따라 요청 별 서비스 강등 적용
- 침해감내 프락시 서버가 이중화되어 구성되는 경우 상호 상태 검사 및 복구
- 이중화된 침해감내 프락시 서버들 사이의 연결 및 침해감내 프락시 서버들과 침해감내 프락시 서버 관리콘솔과의 연결은 서비스 요청, 응답을 위한 네트워크와 분리하여 구성 (out-of-band)
- 서비스 제공자(COTS 서버)에 대하여 복구 및 재 시작 명령 전달

3.1.2 침해감내 프락시 서버 관리콘솔

침해감내 프락시 서버와 out-of-band 로 연결되어 서비스 제공자, 침해감내 프락시 서버 정보를 등록하거나 그룹화하고 서비스 강등 및 유효성 검사 등에 대한 정책을 작성할 수 있는 사용자 인터페이스를 제공하는 시스템으로 다음과 같은 기능을 제공한다.

- 침해감내 프락시 서버 운영정책 (서비스강등, 응답유효성검사, 서비스 제공자 관리, 침해감내 프락시 서버 관리) 편집
- 요청에 따라 침해감내 프락시 서버로 침해감내 프락시 서버 운영 정책 전달
- 침해감내 프락시 서버 및 서비스 제공자(COTS 서버) 운영상태 조회

3.1.3 침해감내 에이전트

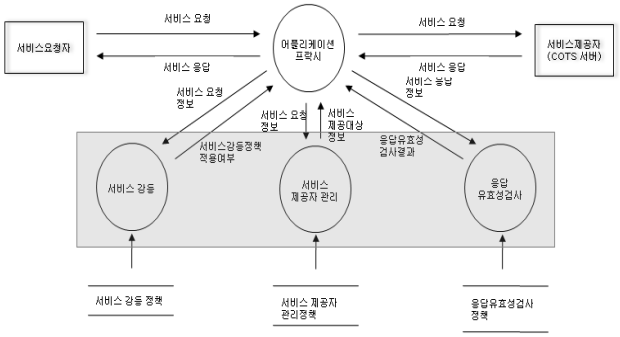
실제 서비스를 제공하고 있는 서비스 제공자(COTS 서버)에 설치되어 운영되며 침해감내 프락시로부터 서비스 중지, 복구, 재 시작 명령 수신하여 수신된 명령에 따라 대상 서비스를 중지, 복구, 재 시작 시킨다.

3.2 동작

침해감내 프락시 서버 프레임워크는 크게 다음과 같은 3 가지의 주요 동작 모드를 가진다.

- 서비스 요청 처리
- 서비스 이주 및 복구
- 서비스 강등

3.2.1 서비스 요청 처리



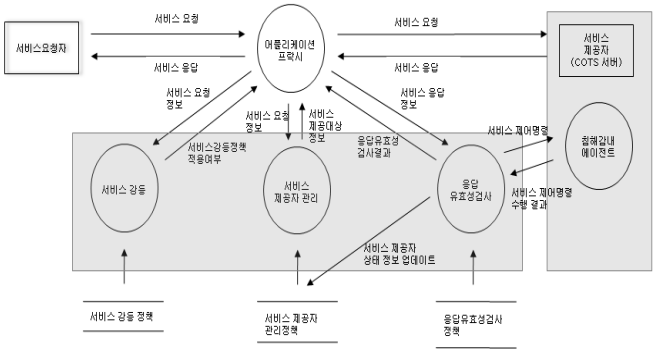
[그림 3] 서비스 요청 처리

침해감내 프락시 서버 프레임워크의 기본적인 동작인 서비스 요청 처리는 다음과 같은 시나리오로 동작한다.

- 서비스 요청자는 제공받으려는 서비스에 대한 서비스 요청을 어플리케이션 프락시로 전달한다.
- 어플리케이션 프락시는 서비스 요청을 수신한다.

- 어플리케이션 프락시는 수신된 서비스 요청에 대한 정보를 입력 값으로 서비스 강등 조회 API 를 호출한다.
- 서비스 요청에 대한 정보를 입력 받은 서비스 강등 조회 API 는 서비스 강등 기능이 활성화된 경우 해당 서비스 요청이 서비스 강등 정책에 적용되는지 여부를 확인한 후 해당 서비스 요청을 서비스 제공자로 전달할 지의 여부를 결과로 반환한다.
- 서비스 강등 조회 API 로부터 해당 서비스 요청에 대한 전달 여부를 전달받은 어플리케이션 프락시는 요청된 서비스에 대한 서비스 제공자 정보를 얻기 위해서 수신된 서비스 요청에 대한 정보를 입력 값으로 서비스 제공자 조회 API 를 호출한다.
- 서비스 요청 정보를 입력 받은 서비스 제공자 조회 API 는 서비스 제공자 관리 정책에 기록된 내용을 읽어 해당 서비스 요청을 전달해야 하는 서비스 제공자 목록 정보를 결과로 반환한다.
- 어플리케이션 프락시는 반환된 서비스 제공자 목록에 있는 동일한 서비스를 제공하는 중복된 서비스 제공자 들에게 동일한 서비스 요청을 전달한다.
- 서비스 제공자(COTS 서버)는 수신된 요청을 처리하고 이에 대한 응답을 어플리케이션 프락시로 전달한다.
- 서비스 제공자(COTS 서버)로부터 응답을 수신한 어플리케이션 프락시는 응답에 대한 정보를 입력 값으로 응답 유효성 검사 API 를 호출한다.
- 응답 유효성 검사 API 는 사전에 정의된 응답 유효성 검사 정책에 따라 수신된 응답의 유효성을 검사하고 유효한 응답인지의 여부를 결과로 반환한다.
- 어플리케이션 프락시는 응답 유효성검사 API 로부터 유효한 응답으로 결과가 반환된 응답을 서비스 요청자로 전달한다.

3.2.1 서비스 이주 및 복구

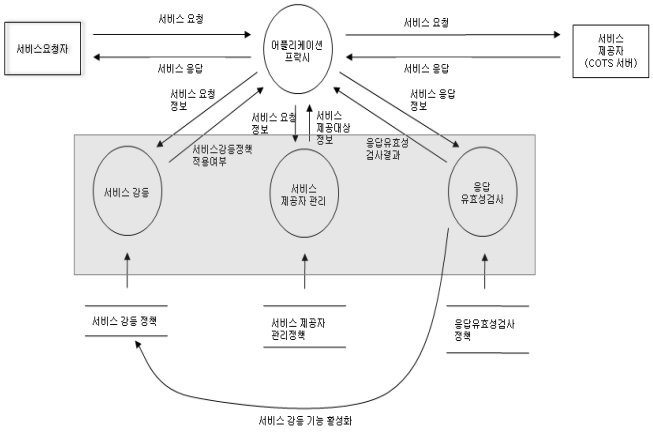


[그림 3] 서비스 이주 및 복구

서비스 이주 및 복구는 응답 유효성 검사를 통하여 대상 서비스에 이상이 발견되는 경우 동일한 서비스를 제공하는 다른 서비스 제공자로 서비스를 이주하고 손상된 서비스를 복구 하기 위한 것으로 다음과 같은 시나리오로 동작한다.

- 서비스 제공자로부터 응답을 수신한 어플리케이션 프락시는 응답에 대한 정보를 입력 값으로 응답 유효성 검사 API 를 호출한다.
- 응답 유효성 검사 API 는 사전에 정의된 정책에 따라 수신된 응답의 유효성을 검사하고 유효한 응답인지의 여부를 반환한다. 이 때 일정기간 동안 유효하지 응답으로 판정된 비율이 사전에 정의된 응답 유효성 검사 정책에 기술된 서비스 제어를 위한 임계 치를 초과하는 경우 해당 서비스 응답을 제공한 서비스 제공자(COTS 서버)에 설치된 침해감내 에이전트로 해당 서비스에 대한 중지 및 복구 명령을 전달한다.
- 서비스에 대한 중지 및 복구 명령을 수신한 침해감내 에이전트는 사전에 정의된 스크립트에 따라서 서비스에 대한 중지 및 복구를 수행한 후 수행결과를 침해감내 프락시 서버 관리콘솔로 보고한다.
- 침해감내 프락시 서버 관리콘솔은 수신된 서비스 제공자의 응답 상태에 대해 서비스 제공자 관리 정책에 기록하여 다음 번 서비스 제공자 요청 API 가 호출되는 경우 서비스가 중지되거나 복구가 실패한 서비스 제공자는 반환 목록에 포함시키지 않도록 한다.

3.2.1 서비스 강등



[그림 3] 서비스 강등 동작구조

서비스 강등은 침해감내 프락시 서버 프레임워크의 핵심적인 동작으로서 네트워크 또는 시스템에 장애가 발생하여 정상적인 서비스가 어려운 상황이 발생하는 경우 사전에 정의된 정책에 따라 우선순위가 높은 요청만을 서비스 제공자로 전달하는 기능이다. 서비스 강등 기능은 다음과 같은 시나리오로 동작한다.

- 침해감내 프락시 서버 운영정책 중 서비스 강등 정책은 서비스 별로 부여된 우선순위에 대한 정보를 저장하고 있다.
- 응답 유효성 검사 API 는 사전에 정의된 정책에 따라 수신된 응답의 유효성을 검사하고 유효한 응답인지의 여부를 반환한다. 이 때 일정기간 동안 유효하지 않은 응답으로 판정된 비율이 사전에 정의된 응답 유효성 검사 정책에 기술된 서비스 강등을 위한 임계 치를 초과하는 경우 응답 유효성 검사 API 는 서비스 강등 기능을 활성화 한다.
- 서비스 요청자는 제공받으려는 서비스에 대한 서비스 요청을 어플리케이션 프락시로 전달한다.
- 어플리케이션 프락시는 서비스 요청을 수신한다.

- 어플리케이션 프락시는 수신된 서비스 요청에 대한 정보를 입력 값으로 서비스 강등 조회 API 를 호출한다.
- 서비스 요청에 대한 정보를 입력 받은 서비스 강등 조회 API 는 서비스 강등 기능이 활성화된 경우 요청된 서비스가 서비스 강등 정책에 기술된 내용에 따라 서비스제공자로 전달되어야 하는 지의 여부를 확인한 후 해당 서비스 요청을 서비스 제공자로 전달할 지의 여부를 결과로 반환한다.
- 서비스 강등 조회 API 로부터 해당 서비스 요청에 대한 전달 여부를 전달받은 어플리케이션 프락시는 전달되지 않아야 하는 서비스 요청으로 판단되는 경우 해당 요청을 서비스 제공자에게 전달하지 않는다.

4. 결론

본 논문에서 제안한 침해감내 프락시 서버 프레임워크는 기존의 시스템 및 네트워크 인프라에 대한 재 구성 및 수정을 최소화하면서 주요 서비스에 대한 생존 성을 강화 시킬 수 있는 구조로 설계되었으며 다양한 어플리케이션에 적용 가능 하도록 어플리케이션에 의존적인 부분들에 대해서는 API 형태의 인터페이스를 제공하여 필요에 따라 구현되어 적용될 수 있도록 하였다.

향후 연구 과제로는, 다양한 어플리케이션에 대한 응답 유효성 검사를 위한 유효성 검사 정책 등의 세부적 사항들을 설계하고 해당 기술을 구현하여 제안된 구조에 대한 검증을 수행할 예정이다.

6. 참고 문헌

[1] Jaynarayan Lala, DAPRA's Path to Self-Regenerative System. June 2002. http://www.laas.fr/IFIPWG/Workshops&Meetings/42/03-Lala.pdf

[2] Deswarte, Y., et.al, "Intrusion Tolerance in Distributed Systems", In IEEE Symp. On Research in Security and Privacy, Oakland, CA, USA, pp.110-121, 1991.

[3] Reynolds, J., et. al, "The Design and Implementation of an Intrusion Tolerant System", Proceedings of the International Conference on Dependable Systems and Networks (DSN'02), 2002.

[4] Wang, F. and Killian, C., "Design and Implementation of SITAR Architecture: A Statue Report", Proc, of the ICDSN 2002 Supplementary Vol., p.C-3-1, June 2002.

[5] Neves, N. F. and Verissimo, P., The Middleware architecture of MAFTIA: A blueprint. DI/FCUL TR 99-6, Department of Computer Science, University of Lisboa, Sept. 2000.

[6] Adelsbach, A., et. al, "Conceptual Model and Architecture of MAFTIA," Project MAFTIA IST-1999-11583 deliverable D21. 2002.

[7] Powell, D., et.al, "MAFTIA (Malicious- and Accidental-Fault Tolerance for Internet Applications)," Sup. of the 2001 International Conference on Dependable Systems and Networks (DSN2001), G oteborg (Sweden), 1-4 July 2001

[8] Alfonso Valdes., et. all Dependable Intrusion Tolerance. International Symposium on Recent Advances in Intrusion Detection (RAID) Oct 2001