분산 컴포넌트 시스템의 보안 강화를 위한 운영체제 보안 모듈의 설계

강진석* · 강흥식**
*인제대학교 정보컴퓨터공학과
comdoll2@netian.com
hskang@nice.inje.ac.kr

Design of the OS Security Module for the Security Enhancement in a Distributed Component System

Jin-Suck Kang* · Heung-Seek Kang**
*Dept of Information and Computer Engineering, Inje University

요약

과거의 프로그램은 단일 프로그램으로 작성될 경우, 작성하기도 어렵고 관리 또한 용이하지가 않았다. 결국, 오늘날에는 이를 해결하고자 큰 프로그램을 작고 이해하기 쉬운 분산 컴포넌트별로 나누는 방식을 활용하고 있다. 하지만, 이러한 분산 컴포넌트 기반 소프트웨어는 보안 측면에서 볼 때 상당히 위험한 요소들을 내포하고 있다. 즉, 외부나 내부에서 독립적으로 링크되는 개개의 컴포넌트들이 보안을 고려한 모든 상황에서 안전하게 이용된다는 보장이 없다. 본 논문에서 제안하는 시스템은 바로 이러한 점을 해결하고자 운영 체제에 보안 모듈을 내장하고 이 보안 모듈로 하여금 개개의 컴포넌트가 링크될 때에 그 안전성을 검증하고 혹시 있을지 모를 불법적인 컴포넌트 조작을 사전에 막을 수 있도록 설계하였다.

1. 서 론

현재 분산 컴포넌트 형태의 응용 프로그램은 개발의 편의성과 생산성 그리고 유지보수의 편리성으로 인해 널리 보급되고 있는 프로그램 개발 방법 중에 하나이다.

이른바, 분산 객체 지향 프로그래밍, DCOM, CORBA, 플러그인 등이 바로 그러한 방법을 대변하는 산물들이라 할 수 있다.

즉, 과거와 같이 운영체제 위에 하나의 단일 응용 프로그램들을 동작시키는 방식과는 달리 여러 응용 프로그램이 공동으로 사용할 수 있는 다양한 컴포넌트 형태의 분산 객체를 동적으로 링크시켜 동작시키는 것이다.

그러나, 이러한 다양한 분산 컴포넌트의 개발과 활용은 오늘날 보안이라는 측면에서 보았을 때 그 위험성이 나날이 증가되는 결과를 초래할 수 있다.

결론적으로 말해, 개발자의 입장에서 편리함만을 추구하여 개발하고자 하는 응용 프로그램에 외부로부터 개발된 기능상으로 신뢰성이 입증되지 않는 컴포넌트를 이식하게 되면 자신이 개발하고자 하는 기본 프로그램과 보이지 않는 부분에서 충돌이 발생할 수 있을 뿐더러 잠재적으로는 그것이 시스템 전체적인 면에서 보안상의 허점으로 작용할 소지가 높다고 할 수 있는 것이다

본 논문에서는 바로 이러한 상이한 분산 컴포넌트의 활용을 통해 발생할 수 있는 보안상의 문제점을 제시하고 이것을 해결할 수 있는 방법으로 운영체제상의 보안 모듈을 설계하고자 한다.

먼저, 2장에서는 현재 국내에서 추진중인 분산 컴포넌트 개발의 현황을 살펴보고, 3장에서는 보안 측면에서 이러한 분산 컴포넌트가 유발할 수 있는 문제점을 짚어본다. 그리고 본격적으로 4장과 5장에 이어서는 본 논문에서 제시하는 운영체제 보안 모듈의 구성과 설계 그리고 그 기능상의 동작형태들을

서술할 것이다

끝으로 6장에서는 본 연구의 결론을 통해 차후 본 연구의 발전을 위한 제반 사항에 대해서 언급하고 논문을 맺도록 하겠다.

2. 분산 컴포넌트 개발의 현황

현재 우리나라 IT 업계에서는 분산 컴포넌트가 21세기 정보 기술 산업의 새로운 패러다임으로 급부상할 것으로 보고 있다.

사실, 개발자의 입장에서 보아 분산 컴포넌트의 개발과 활용은 IT 산업의 Life-Cycle의 단축으로 인해 짧은 시간 안에 필요한 응용 프로그램을 개발해야 하는 부담을 크게 줄여주고 있다.

이렇듯, 기존에 개발되어 있는 여러 컴포넌트들을 원하는 형태로 재 조합하여 새로운 응용 프로그램을 개발한다는 것은 비용의 절감과 생산성 향상이라는 이득과 맞물려 IT 산업 전반의 새로운 개발 형태로 자리 매김 할 것으로 보여진다. 이를 반증이라도 하듯 우리나라에서 컴포넌트 사업을 추진하는 업체들도 지난해 20~30개 수준에서 100여 개로 크게 늘고 있다. 뿐만 아니라 이러한 환경 변화에 발맞추어 정보통신부에서는 2002년 올해까지 1900개의 컴포넌트를 개발 육성하겠다는 계획을 추진중이다[1].

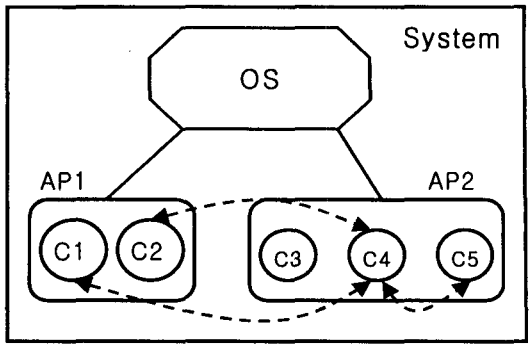
3. 분산 컴포넌트의 보안상 문제점

오늘날, 응용 프로그램을 개발하는 프로그래밍 기법은 운영체제 위에 하나의 단일 프로그램을 동작시키는 종래의 방법을 탈피하여 프로그램의 크기를 줄이고 동시에 개발과 유지 보수의 편의성을 도모하기 위해 점차 분산 컴포넌트를 활용한 재 조합 과정을 통해서 새로운 프로그램을 개발하는 방향으로 변모하고 있다[3].

[그림1]은 이러한 분산 컴포넌트를 통해 개발된 응용 프로그램이 운영체제 상에서 동작하는 형태를 보여주고 있다.

그림에서는 두 개의 응용 프로그램(AP1과 AP2) 안에 각각 C1~C5로 표시된 컴포넌트들이 내장된 형태로 구성되어 있다. 그리고, 각각의 컴포넌트들은 플러그인 형태로 제각기 별도의 기능을 제공하고 있을 뿐만 아니라 경우에 따라서는 컴포넌트끼리도 운영체제의 영향을 받지 않고 동적 링크를 통해서 서로 통신할 수 있게 되어 있다.

이렇듯, 오늘날 컴포넌트는 보통 응용 프로그램을 실행하는 시점에서 DLL(Dynamic Linking Library) 형태로 링크된다.



[그림 1] 분산 컴포넌트 기반의 응용프로그램 형태

바로 이러한 부분에서 보안 문제는 여러 방향으로 발생할 수 있다. 과거 단일 형태로 만들어진 응용 프로그램들은 개발자가 응용 프로그램에 필요한 모든 파일 조각들을 공유 라이브러리(Shared Library)로 합치는 링킹(Linking) 과정을 통해 통합한 후에 모든 요소가 제대로 안전하게 동작하는지를 확인할 수 있었다. 그러나 앞서 말한 분산 컴포넌트 기법을 활용한 응용 프로그램의 경우는 사실상 그러한 과정이 불가능하다.

이러한 이유에서 분산 컴포넌트 형태의 개발은 보안에 관련하여 다음과 같은 문제점들을 야기할 수 있다.

첫째. 각각의 컴포넌트 모두를 절대적으로 신뢰할 수가 없다. 이것은 컴포넌트들이 독립적으로 분산되어 있기 때문에, 해당 시스템이 해킹을 당할 경우 컴포넌트 자체가 변조 및 다른 형태로 위조될 가능성이 있다는 것을 의미한다.

둘째, 모든 컴포넌트들이 가능한 모든 상황에서 제대로 작동하도록 설계되었다고 보장할 수 없다. 물론, 컴포넌트 개발자들은 표준화된 형태의 프레임워크와 규칙을 기반으로 제작되었다고 주장할 수 있다. 그러나 만일 개발에 사용되는 툴 들이 동일하지 않는 조건에서 그 모든 컴포넌트들이 어떠한 상황에서도 완벽하게 조합될 것이라고 믿는 것은 큰 오산이 아닐 수 없다. 더구나, 이러한 문제점은 잠재적으로 해킹에 있어서 결정적인 취약점으로 나타날 가능성이 무척 높다.

셋째, 현재의 운영체제로는 위의 두 가지 문제를 해결할 방법이 없다는 것이다. 과거에는 공유 라이브러리들이 운영체제를 통해 상호 통신을 하였다.

따라서 제대로 된 운영체제의 경우, 이러한 통신에 개입하여 프로그램끼리 피해를 주지 않도록 하는 것이 어느 정도는 가능하였다. 그러나, 분산 컴포넌트의 경우는 [그림 1]에서와 같이 운영체제로부터 독립적으로 동작하기 때문에 이와 같은 기능이 적용될 수 없다.

4. 운영체제 보안 모듈의 구성

이러한 보안상 안전성의 문제를 해결하기 위해 본 논문에서는 운영체제에 보안 기능을 수행할 수 있는 컴포넌트 보안 모듈을 내장하는 시스템을 제안한다.

이 시스템은 다음의 두 가지 원칙을 기반으로 하고 있다.

① 컴포넌트별 개별 코드 서명

- 운영체제는 자신의 시스템에 동작중인 여러 컴포넌트들을 설치 시 미리 개별 코드 서명을 부여하여 인터넷이나 외부로부터 검증 받지 않는 컴포넌트가 유입될 경우 이를 제한할 수 있는 기능을 가진다.

현재 우리가 가장 많이 활용하고 있는 인터넷의 경우 사용자가 감지하지 못하는 사이 원격지에서는 다양한 형태의 컴포넌트들이 플러그 인 형태로 사용자 브라우저를 통해서 유입된다. 보통의 경우 일반적인 사용자는 그 컴포넌트들의 무결성을 검증하지 않고 바로 설치하고 사용하는 경향이 높다. 이 때에 그 컴포넌트가 해킹을 위한 백도어의 기능을 지닌 것이라면 해당 시스템은 치명적인 결과를 가져올 수 있게 된다.

② 할당된 메모리 영역을 통한 컴포넌트의 활동제한

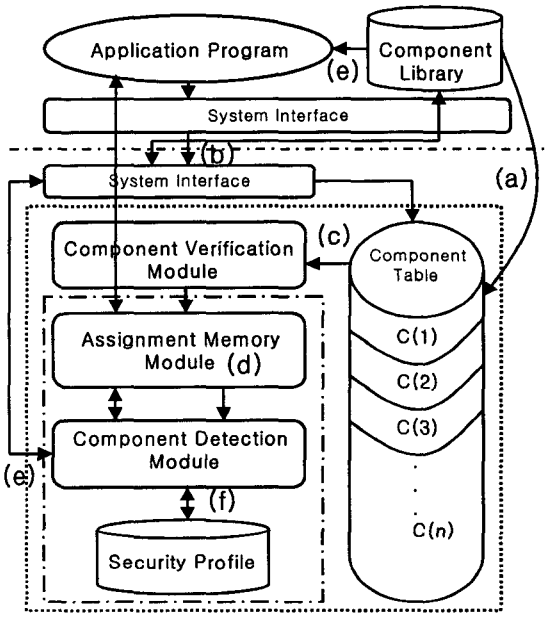
- 이 원칙은 악의적인 컴포넌트에 의해 다른 컴포넌트나 시스템 전반에 끼칠 수 있는 영향을 차단한다. 다시 말해 컴포넌트에 고유의 메모리 영역을 할당함으로써 다른 컴포넌트의 영역을 침범하거나 변경하는 것을 막을 수가 있고, 필요에 따라서는 컴포넌트를 감시하는 모듈이 부적절한 동작을 사전에 방지할 수가 있다.

[그림 2]는 운영체제 내부에서 동작하는 보안 모듈과 응용 프로그램의 구성과 연동 형태를 보여주는 그림이다.

우선, 컴포넌트는 시스템 내부로 들어온 후 운영체제 보안 모듈의 Component Table에 코드 서명 과정을 통한 컴포넌트 등록이 이루어진 상태이다(a). 그 후 해당 응용 프로그램이 Component Library로부터 원하는 컴포넌트를 링크하기를 요청하게 되면 (b) Component Verification Module은 Component Table에 등록된 코드 서명에 근거하여 현재 요청한 컴포넌트가 등록 이후 아무런 변조나 위조가 없었음을 검증하게 된다(c).

검증과정을 마치고 나면 다음으로 Assignment Memory Module은 해당 컴포넌트가 동작할 메모리 공간을 할당하고 관리한다(d). 이것은 컴포넌트가 접근할 수 있는 자원과 범위를 제한함으로써 만일의 경우 발생할 수 있는 컴포넌트 변조를 사전을 막을 수 있을 뿐만 아니라 다른 컴포넌트에게 끼칠 수 있는 영향을 최소화하는 효과를 얻을 수 있다.

마지막으로 Component Detection Module에서는 내장된 Link Management를 통해서 해당 응용 프로그램이 요청한 컴포넌트를 링크시키거나 이상 징후가 발견될 경우에는 컴포넌트의 링크를 막는 역할을 한다(e).



[그림 2] 운영체제 보안 모듈의 구성도

5. 보안 모듈내의 Component Detection Module의 구조

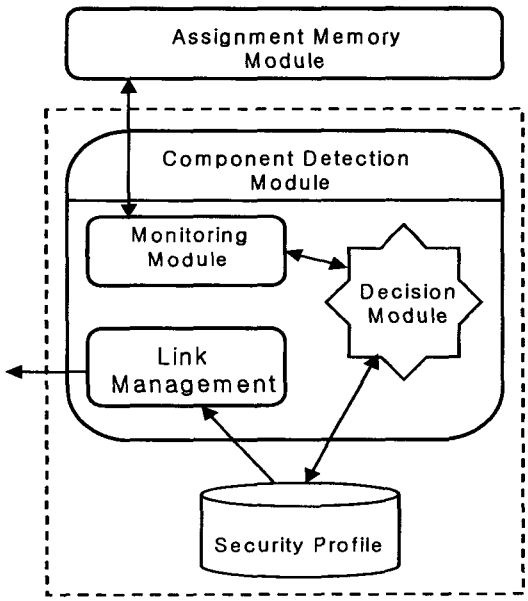
Component Detection Module은 컴포넌트별로 메모리를 할당하고 관리하는 Assignment Memory Module로부터 현재 동작중인 메모리 영역의 컴포넌트 정보를 수신하여 감시하는 Monitoring Module과 링크를 원하는 응용 프로그램에게 해당 컴포넌트를 링크시키거나 차단시키는 Link Management Module로 구성되어 있다.

이 때 할당된 메모리 상에서 동작중인 컴포넌트에게 발생할지도 모를 불법적인 변조나 이상 징후를 탐지하는 방식은 다음과 같다.

① Monitoring Module은 Assignment Memory Module로부터 현재 할당된 메모리 상에서 동작중인 컴포넌트의 영역의 자원과 범위에 관한 정보를 수신한다.

② 이 때 만일 이상 징후에 관한 정보를 수신하게 되면 즉시 Monitoring Module은 Decision Module에게 현재 상황에 관련한 보안 정책 사항을 요청하게 된다.

③ Decision Module은 컴포넌트의 자원에 관한 행동 범위와 자원에 관련된 정보가 정책적으로 구성되어 있는 Security Profile에게 문의하여 컴포넌트 이상 유무에 관한 판단을 내린다. 만일 여기서 이상 사항이 판단되면 즉시 Link Management Module은 링크된 컴포넌트를 차단시킨다.



[그림 3] Component Detection Module의 구조

6. 결론 및 향후 연구 과제

이상 서술한 바에 의해 본 논문에서 제안하는 시스템은 앞서 말한 원칙에 따라 응용 프로그램의 컴포넌트 활용에 있어서 두 가지의 보안 조건을 만족시킨다.

첫째, 응용 프로그램의 설치나 인터넷을 통해 외부로부터 유입되는 각종 컴포넌트들을 코드 서명에 근거한 검증 과정을 수행함으로써 컴포넌트들의 변조나 위조에 따른 악의적인 해킹으로부터 시스템을 사전에 보호할 수 있다.

둘째, 컴포넌트가 활동하는 자원과 범위를 메모리 상에서 제한함으로써 해킹에 의한 컴포넌트의 시스템 내부 자원에 대한 접근을 원천적으로 차단시킬 수 있다. 동시에 컴포넌트가 불법적으로 변조되거나 주어진 자원에서 벗어난 행위를 할 경우 이를 감시할 수 있는 여건을 제공해 준다.

본 논문은 분산 컴포넌트의 검증 과정을 통한 신뢰성 보장과 컴포넌트 활용에 있어서 발생할 수 있는 변조 및 해킹에 대한 개연성을 사전에 예방하는 방법을 제안한 것이다. 특히, 오늘날 IT 산업의 새로운 사업으로 급부상하고 있는 컴포넌트 시장이 몰고 올 시스템 보안의 새로운 파장을 언급하고 이를 해결할 수 있는 대안적 방안을 운영체제 내의 보안 모듈을 통해 설계하였다.

향후, 이 시스템이 새로운 소프트웨어 시장의 변화에 발맞추어 발전하기 위해서는 컴포넌트 개발에 있어서 보안 사항을 준수할 수 있는 완벽한 프레임워크의 표준화가 필요할 것이다. 그리고 아울러 현재 기능적인 면에만 편중된 소프트웨어 개발자들에게는 앞으로 자신이 개발할 컴포넌트들이 해당 플랫폼의 어떤 부분에서 보안적 취약점을 드러낼 수 있는가에 대한 무게 있는 고찰이 선행되어야할 필요성이 있다.

7. 참고 문헌

[1] http://minerva.imeca.co.kr/news/news2_4.html.

[2] http://comedu.skku.ac.kr/~kcn1209/security.html.

[3] http://www.component.or.kr/

[4] 김용권 역, "해커프루프", 정보문화사, 1998.

[5] 방효찬,김명은,장종수, "PBNM 최신 동향 분석을 통한 정책기반의 네트워크보안제어 기술 제안", NCS2000.

[6] 허영준 외, "보안정책모델을 적용한 Security Policy Agent 구조", COMSW2001.

[7] M. Bishop, "A Standard Audit Trail Format. In Proceeding of the 18th National Information System Security Conference", Baltimore.

[8] J,G.Ko, S. Y. Doo, S.K. Un and J.N. Kim, "Design and Implementation for Secure OS Based on Linux", Proceedings of WISA2000.