經營科學
第17卷 第2號
2000年 11月

클라이언트-서버 시스템에 대한 통제 : 은행업체에 대한 사례연구

이 상 재*

The Controls of Client-Server System : Case Studies of Banks

Sang-Jae Lee*

Abstract

The introduction of client server systems raises some serious IS control concerns. Although client server systems increase system flexibility and compatibility, and facilitates system downsizing and process restructuring, multiple operating systems, database management systems, and platforms of client-server system increase the potential exposures and risks of the system. It is needed to recommend the effective control framework to mitigate risks. This paper suggests risks and control framework for client server system. The results of two case studies indicate that program and database management are important as program and data should be located and maintained in client and servers. It becomes difficult to manage divided programs and data logic that are distributed across clients and servers that have different platforms. The case studies of two banks show that the extent of distribution of program and data logic affects the importance of program and database management controls.

1. 서 론

클라이언트-서버 시스템(client-server)은 네트워크를 통해 연결된 컴퓨터간에 업무를 분산시키고 정보를 공유시키는 분산화된 처리시스템이다. 클라이언트는 서비스를 요청하고 서버는 이러한 서비스 요청을 수신하여 처리한다(Held, 1994). 클라이언트-서버 시스템은 중앙집중식 시스템에 비

* 한양대학교 디지털 경영학부 교수

해서 유연성을 가지고 업무 생산성을 향상시키기 때문에 사용자 만족을 증가시키며 최근 급속히 보편화되어가고 있다(Anandarajan and Arinze, 1998). 1992년에 시행된 설문조사의 결과 미국기업은 전체 응용시스템 중에 55-60% 정도는 클라이언트-서버 시스템 방식으로 구성되어 있고 이러한 시스템으로 구축되는 응용시스템의 수는 계속적으로 증가하고 있다. 클라이언트-서버 시스템은 사용자가 보유하고 있는 네트워크상의 워크스테이션만큼의 데이터 엑세스(access) 지점들을 제공한다. 이것은 사용자가 좀더 많은 도구들을 사용하여 데이터를 처리할 수 있도록 해준다. 클라이언트-서버 시스템의 시스템 구성은 사용자의 요구가 발전함에 따라 유연하게 변화될 수 있다. 즉 사용자가 자신의 독특한 응용프로그램에 맞도록 시스템을 조정할 수 있으며 자신의 작업에 비용면에서 가장 효율적인 하드웨어와 소프트웨어를 구성할 수 있다(Atre, 1994).

분산된 시스템에서는 중앙집중식 시스템에 비해서 보안상의 위험이 대체로 증가한다고 지적되고 있다. 관리자들은 전통적인 메인프레임을 이용한 중앙집중식 시스템보다도 클라이언트-서버 시스템의 위험이 증가하는 것을 제대로 인식하지 못하고 이에 대한 관리를 소홀히 할 수가 있다(Cairo and Friedberg, 1995 ; Wood, 1994). 실제로 최근에 행하여진 설문연구조사의 결과에 의하면 기업들은 중앙집중식 시스템에 비하여 클라이언트-서버 시스템에서 보안위험에 대비한 적절한 통제대책을 수립하지 못하고 있다(Ryan and Bordoloi, 1997). 클라이언트-서버 시스템은 다양한 공급자(vendor)들이 제공한 플랫폼(platform), 운영체제, 프로토콜 및 응용시스템으로 구성되어 업무 처리의 유연성 및 생산성이 높아졌지만 이러한 복잡한 플랫폼에서 발생하는 오류를 발견하고 해결하기가 더욱 어려워졌다. 또한 데이터 및 응용프로그램에 불법적으로 접근할 수 있는 경로와 가능성이 높아졌다.

클라이언트-서버 시스템에서 보안위험은 대부분 데이터나 시스템에의 부적절하고 비합법적인 접근(예 : 비합법적인 로그-온 발생위험, 외부로부터 데이터 및 시스템에의 접근, 접근자에 의한 부적절한 권한 설정, 데이터에 대한 변조 및 삭제)과 연관된다. 일반 사용자들이 네트워크를 통해서 원하는 데이터를 사용하고 스스로 작업을 처리하므로 오류나 불법적인 데이터 사용 및 작업의 가능성이 커진다. 응용시스템 유지관리업무도 전적으로 사용자에게 의존하는 경우가 많으므로 응용프로그램 변경 및 개발과정에 적절한 표준 및 절차를 수립하지 않으면 프로그램의 무결성, 호환성, 안전성에 문제가 발생할 수 있다. 클라이언트-서버 시스템은 업무처리능력 및 전산자원을 다양한 클라이언트에 분산시킴으로써 유연성을 향상시켰지만 개방된 전산환경으로 인해 다양한 지점 및 경로를 통한 불법적 접근에 노출되게 되었다. 다양한 하드웨어나 소프트웨어로 구성된 클라이언트-서버 시스템에서는 보안정책을 표준화하고 일관되게 적용시키는데 어려움이 있다(Ryan and Bordoloi, 1997). 예를 들면 시스템사용자의 업무처리를 승인 및 확인하는 접근통제 방법도 특정 하드웨어나 소프트웨어 제품마다 다르므로 한가지 방법으로 전체 기업정보시스템에 적용시키기가 힘들다. 클라이언트-서버 시스템의 효과성 및 안전성을 위해서 이러한 위험을 분석하고 통제방안을 제시하는 것이 필요하다.

클라이언트-서버 시스템은 데이터의 분산화나 처리의 분산화 정도에 따라서 여러 아키텍처(architecture)가 존재한다(Chengalur-Smith and Duchessi, 1999). 본 논문은 두 가지 클라이언트-서버 아키텍처(로직 (logic)분할, 원격데이터) 에 대한 위험과 통제방안을 구체적인 사례를 들어서 제시하여 보았다. 사례분석을 통해서 실제로 클라이언트-서버 아키텍처의 특성때문에 생길 수 있는 위험이 존재하며 이에 따른 통제방안을 제시할 필요가 있음을 설명하였다. 2장에서는 클라이언트-서버 시스템의 두가지 구조인 로직 분할 및 원격 데이터구조에서 응용시스템 개발통제, 데이터베이스관리통제가 중요함을 연구모형으로 제시하였다. 본 논문의 초점

은 사례연구를 통해서 응용시스템관리에 관련된 위험, 데이터베이스관리에 관련된 위험이 클라이언트-서버 시스템 구현으로 인하여 증가함을 보여주고자 하는데 있다.

2 이론적인 배경

2.1 클라이언트-서버의 보안문제

정보시스템 통제는 조직통제의 한 형태로 볼 수 있는데 결국 자산을 보호하고 데이터의 무결성을 유지하고 조직목표를 효과적으로 달성하며 자원을 효율적으로 소비하기 위해 지원되는 활동이다(Weber 1999). 사실 정보시스템 통제의 범위는 상당히 광범위해서 조직의 효과성 및 효율성에 영향을 미치고 조직이 정보시스템 구현을 통해서 조직목표를 달성하고자 수행하는 모든 활동 예를 들면 전략계획이나 정보시스템 관리활동이 정보시스템 통제의 범위에 포함될 수 있다. 본 논문에서는 기존의 보안통제에 대한 연구가 거의 없기 때문에 정보시스템통제의 네 가지의 목표 중에 보안 및 무결성을 확보 및 증진하는 활동을 통제라고 정의하고 이 부분에 대해 연구하고자 한다. 정보시스템 보안은 정보시스템 기밀성, 무결성, 가용성의 세가지 속성으로 구성된다. 결국 정보시스템 보안 통제는 시스템의 기밀성, 무결성, 가용성의 향상을 위해서 취해지는 절차 및 행위라고 정의한다. 이는 궁극적으로 조직이 정보시스템의 구축을 통해서 조직의 목표를 달성하고자 하기위해 취해지는 것이라고 볼 수 있다(Weber 1999). 보안위험은 정보시스템 보안통제가 소기의 목적을 달성하지 못하여 내외부의 각종 고의적이거나 비고의적인 위협(예 : 자연재해, 해커의 공격, 업무오류, 시스템 에러)의 발생으로 인하여 정보시스템의 손실이 발생하는 것을 의미한다.

클라이언트-서버 시스템의 보안은 복수의 운영체제, 데이터베이스 관리시스템, 응용시스템 및 데이터 등의 보안과 관련된다. 클라이언트-서버 시스템에서는 네트워크, 컴퓨터 등의 구성 요소 중 한 요소의 위험이 다른 요소에 위험을 준다. 예를 들면 네트워크의 여러 지점에서부터 오는 접근 시도에 대한 인증이 적절히 이루어지지 않을 경우 시스템 전체의 보안성에 위협을 가져올 수 있다(Hancock, 1994 ; Symonds, 1994). 원격에 있는 터미널에서 클라이언트 워크스테이션에 대한 접근에 대한 통제, 서버 로그 인에 대한 통제, 불법적인 다이얼-업 접근에 대한 통제가 중요하다(Lyons, 1994). 또한 분산처리시스템에서는 종이로 된 감사증적을 유지하는 것이 불가능하기 때문에 전자감사증적에 대한 관리통제 및 감사가 필요하다(Moeller, 1989).

클라이언트-서버 시스템에서는 중앙집중식 시스템에 비하여 사용자에 대한 통제가 더욱 중요해진다(Ryan and Bordoloi, 1997). 클라이언트-서버 시스템의 사용이 증가하면 전산지식이 풍부한 사용자의 수가 대체로 증가하게 되는데 시스템 보안이 시스템에 대한 높은 수준의 활용능력을 갖춘 이러한 사용자에 의해 위협 받을 가능성이 커진다. 시스템의 보안은 결국 분산되어 있는 사용자에 대한 통제를 효과적으로 하는데 의존한다. 사용자접근에 대한 인증뿐만 아니라 사용요금부과 및 오류추적을 위해 거래기록(logging)을 유지해야 한다. 클라이언트는 각각 복수의 서버와 동시에 통신하는데 이 과정에서 클라이언트는 TCP/IP, SNA, IPX와 같은 서로 다른 프로토콜을 수용할 수 있어야 한다(Molini, 1994). 보안통제는 시스템 경계를 넘어서 다양한 사용자에 대해 적용되어지고 추적가능해야 한다. 즉 사용자가 어떤 지점에서 접근을 시도를 할 때 해당 클라이언트와 독립적으로 그를 확인하는 것이 필요하다.

2.2 클라이언트-서버아키텍쳐에 대한 통제

클라이언트-서버 아키텍처는 크게 클라이언트, 서버, 네트워크, 클라이언트-서버용 응용소프트웨어의 네 가지 부분으로 구성되어 있다고 볼 수 있

다. 클라이언트-서버 아키텍쳐는 분산처리를 통해서 응용요소 들이 둘이상의 컴퓨터시스템에 분산되어 구성된다. 표현로직(Presentation Logic), 업무로직(Business Logic), 데이터로직(Data Logic) 등의 분산정도에 따라 아키텍쳐가 달라지게 된다. 클라이언트-서버의 구성 응용로직 들을 살펴보면 다음과 같다(Edelstein, 1994).

- 표현서비스 : 사용자와 접촉하는 부분으로서 표현로직이 의미하는 바를 사용자에게 전달해주는 물리적인 도구(Monitor, Printer, …)들에 의한 행위나 사용자의 입력내용을 접수하는 장치들 (Keyboard, Mouse, Reader, …)이다.
- 표현로직 : 사용자에게 메뉴화면을 제시하고 어떤 메뉴옵션을 선택했을 때 열거된 작업 리스트로부터 원하는 것을 고르게 하고 이때 일어나는 현상을 조정하는 로직이다. 따라서 사용자에게 논리적인 접근통제의 역활을 수행할 수 있는 로직이다.
- 업무로직 : 응용시스템의 구현목적을 달성하기 위한 계산, 의사결정 등의 역활을 수행하는 로직으로서 예를 들면 종업원 급여계산이나 대부여부 평가 등의 업무처리를 수행하는 부분이다.
- 데이터 로직 : 업무로직이 목적을 달성하기 위해서 데이터를 조정 및 통제하는 부분이다. 예를 들면 데이터베이스에서 SQL 즉 "SELECT", "UPDATE", "INSERT"과 같은 것이다.
- 데이터 서비스 : 데이터 조작, 데이터 정의, 거래처리, 반환(Commit, Rollback)등과 같은 DBMS의 행위를 말한다.
- 파일서비스 : DBMS의 기능을 지원하기 위해서 필요한 파일을 데이터베이스에서 찾고 작업영역으로 가져오는 운영차원의 역활을 말한다.

입출력을 담당하는 표현로직(presentation logic)은 사용자와의 인터페이스 부분으로서 대부분 PC나 워크스테이션급인 클라이언트에 배치한다. 본 논문에서는 위의 클라이언트-서버구성요소에 따라서 로직분할 및 원격데이터구조를 제시한다. 클라이언트-서버 아키텍쳐는 이 두 가지 아키텍쳐외에 파일서버(File Server), 원격 프리젠테이션(Remote-Presentation), 3단계구조(Three Tiered) 등이 있으나 본 논문에서는 가장 일반적으로 활용되어지는 이 두 가지 아키텍쳐에 촛점을 맞춘다. 이 두가지 아키텍쳐는 가트너 그룹이 분류한 클라이언트-서버 시스템 아키텍쳐 중에 '분산된 처리로직'(Distributed Processing Logic)과 '지역 처리로직'(Local Processing Logic)에 해당된다(Gartner, 1994).

2.2.1 로직분할 (Split-Logic)

표현서비스를 제공하는 클라이언트와 데이터 서비스를 제공하는 서버에 업무로직과 데이터로직을 적절히 분배하는 구조로서 업무처리 특성이나 하드웨어, 데이터베이스 등과 같은 자원들의 지원정도에 따라서 유연하게 분배할 수 있다.

2.2.2 원격데이터 (Remote-Presentation)

서버는 데이터서비스와 파일서비스 역활을 하고 표현로직과 서비스, 응용로직 들은 클라이언트에서 이루어지는 구조로서 가장 보편적으로 이용된다. 업무로직과 데이터로직이 클라이언트에 존재한다.

<표 1>에 원격데이터 구조와 로직분할 구조의 로직구성을 중앙집중식 시스템과 대비시켜 제시하였다.

메인프레임을 이용한 중앙집중식 시스템과 비교하여 서버 및 클라이언트에 상주하는 응용시스템 및 데이터의 분포가 다르므로 로직분할 및 원격데이터 구조에 대한 통제는 시스템 구성부분 (네트워크, 단말기, 데이터 등)별로 중요성이 달라질 것이다. 예를 들어 네트워크에 대한 부하가 많으면 네트워크에 대한 통제가 다른 통제보다 필요할 것이고 클라이언트에 보다 많이 응용시스템이 분할되어 있으면 클라이언트쪽의 데이터베이스관리나 프리젠테이션통제가 중요할 것이다(Ryan and Bordoloi,

〈표 1〉 메인프레임 및 클라이언트-서버 아키텍쳐(로직분할, 원격데이터)의 특징

	메인프레임을 이용한 중앙집중식 시스템	로직분할	원격 데이터
클라이언트 기능	프리젠테이션 서비스	프리젠테이션 서비스 프리젠테이션 로직 업무로직 데이터 로직	프리젠테이션 서비스 프리젠테이션 로직 업무로직 데이터 로직
서버기능	프리젠테이션 로직 업무로직 데이터 로직 데이터 서비스 파일 서비스	데이터 서비스 파일 서비스 업무로직 데이터 로직	데이터 서비스 파일 서비스
특 징	클라이언트측의 GUI(Graphical User Interface) 기능부족 파일단위로 데이터 전송 메인프레임에 대한 부하큼	클라이언트와 서버에 업무처리 분산 네트워크 전송량이 큼 응용시스템 개발, 유지관리가 어려움	클라이언트의 작업량이 많음 네트워크 부하 (데이터 전송량)가 큼 분산된 데이터에 대한 관리어려움

1997).

중앙집중식 시스템과 비교하여 로직분할 구조와 원격데이터 구조에서 통제의 필요성을 설명하면 다음과 같다 :

● 응용시스템관리

응용프로그램이 서버에 집중되어 있는 경우에 응용프로그램의 유지, 보수가 쉽게 관리될 수 있다. 로직분할이나 원격데이터 구조는 응용프로그램이 클라이언트측에 분산되어 있으므로 메인프레임에 응용시스템이 집중되어 있는 중앙집중식 시스템에 비하여 상대적으로 응용시스템에서 발생되는 문제의 해결이나 관리가 어렵다.

● 데이터베이스관리

네트워크에 전송되는 데이터양은 클라이언트쪽의 업무처리가 많은 경우에 많아진다. 그리고 파일단위로 데이터가 압축되지 않고 전송되는 경우나 서버의 회신이 요약되지 않고 전송되는 경우에도 데이터의 전송양이 많아진다. 네트워크에 전송되는 데이터의 양에 비례하여 전송중의 데이터에 대한 접근, 및 오용의 위험성이 커진다. 대체로 데이터가 서버에서 관리되면 데이터의 일관성(Consistence), 투명성(Tranparence), 무결성(Integrity) 등의 속성을 유지하기가 수월하다. 그런데 로직분할 및 원격데이터 아키텍쳐는 대체로 서버로부터 클라이언트에 데이터가 전송된 후에 업무수행이 이루어지므로 분산되어 처리되는 데이터에 대한 관리통제가 중요하다(Duncan and Sidwell, 1994).

응용시스템이나 데이터가 클라이언트에 분산되면서 사용자에 의해 이루어지는 응용시스템 관리 및 데이터 접근, 조회에 대한 통제의 중요성이 증가한다. 결론적으로 로직분할 및 원격데이터 구조에서는 응용시스템 개발통제, 데이터베이스관리통제가 중요해질 것이다. 이러한 이론적인 배경을 근거로 해서 사례분석을 실시하여 실제로 클라이언트-서버 아키텍쳐의 특성때문에 생길 수 있는 위험을 분석하고 통제방안을 제시해본다.

3. 사례연구의 배경 및 설계

사례연구방법은 연구변수나 가설을 새롭게 탐색하고 몇 가지 대상실체에 대한 심층적인 연구를 위해서 사용된다(Yin, 1993). 규모나 업무의 성격상 보안에 민감하고 클라이언트-서버 환경으로의 이전한 기업사례가 많지않아 대규모의 샘플을 이용한 통계적인 분석방법은 적용하기가 힘들다. 따

라서 클라이언트-서버 시스템의 통제에 대한 심층적인 연구를 위해서 사례연구방식을 연구방법으로 선택하였다. 규모나 업무의 성격상 보안에 민감한 금융산업에 속하고 성공적으로 클라이언트-서버 시스템 환경을 구축한 A, B은행을 사례대상기업으로 선택하였다. A, B은행의 정보계시스템은 각각 로직 분할 및 원격데이터 구조로 구현된 클라이언트-서버 시스템이고 각 은행 정보계 시스템의 통제 및 보안위험이 가지는 특성을 분석하였다. 연구방법은 A은행은 전산부서직원을 대상으로 면담분석방식을 사용하였고 B은행은 면담 및 설문조사방식을 사용하였다. A은행은 현재 로직분할의 클라이언트-서버 시스템을 개발한 경력이 길지 않고 클라이언트-서버 시스템에 대한 대규모 설문조사에 응답자들이 응답하기에 경험이 부족하다고 생각되었고 B 은행은 상대적으로 클라이언트-서버 시스템에 대한 개발 및 운영경험이 있어 응답자들이 설문에 응답하기가 수월하다고 생각되어 면담 및 설문조사방식을 병행하여 적용하였다.

A, B은행은 클라이언트-서버 시스템 구축을 위한 검토에 착수하여 개발을 수행하였다. 이러한 작업을 추진하게 된 배경에는 기존 시스템의 문제점들을 파악한 것이 계기가 되었다. 그러한 기존 시스템의 문제점들은 다음과 같다. 첫째, 시스템규모의 확대 시 기존 시스템을 폐기해야 하므로 전산투자비용이 과다하였다. 둘째, 중요한 소프트웨어를 전산화하기 힘들고 공급업체에 예속되는 경향이 있어 보다 많은 비용을 지불해야 하였다. 셋째, 시스템의 운영에 많은 전산인력을 필요로 하며 유지보수비용이 투자비용에 비례 되어 확대되었다. 넷째, 중앙 집중식 처리방식에 의한 금융시스템에서는 장애발생이 전지역, 전지점의 마비로 직결되어 고객은 물론 사회적 기회비용손실이 증가하였다.

A, B은행은 서버의 운영체제로 유닉스(UNIX) 운영체제를 이용하여 지역분산 및 다운사이징을 구현하였고 네트워크프로토콜로는 TCP/IP를 사용하고 있다. 또한 관계형 데이터베이스관리시스템으로서 오라클(Oracle)을 사용하고 있다. 클라이언트측의 처리프로세스가 지점(Branch) 서버, 지역서버, 그리고 본점의 기능별 서버상의 서버프로세스와 메세지패싱 방식으로 연결되는 구조로 구성되어 있다. 이러한 시스템의 구축으로 인하여 하드웨어벤더의 종속에서 탈피하게 되었고 소프트웨어 개발의 생산성이 향상되었으며 시스템 규모 확대 시 투자 비용 최소화로 인해 전산투자비용의 절감 효과가 있었다.

본 연구에서 언급한 클라이언트-서버 시스템 아키텍처가 위험 및 통제구현에 미치는 영향을 분석하기 위하여 전산팀장 및 직원을 대상으로 면담을 실시하였다. 이를 통해서 통제시스템의 어느 부분이 구현되었는지 파악하였다. 면담 및 설문조사에 사용되는 통제체크리스트를 기존의 일반 정보시스템 안전지침에 관한 문헌을(Bruce, 1995 ; Dykman, 1992 ; Hale, 1996 ; Lainhart and Donahue, 1992 ; Moeller, 1989 ; Weber, 1999) 참고하여 작성하였다(부록참조). 각 세부항목에 대하여 5점 척도로 통제의 수행 및 수립정도를 측정하였다. 통제체크리스트는 응용시스템관리, 데이터베이스관리의 두 가지 분야로 구성되어 있다. 응용시스템관리통제, 데이터베이스관리통제의 연구변수에 대해서 각각 8개의 측정항목을 사용하였다. 응용시스템관리통제는 시스템 개발, 유지, 보수방법 및 절차 및 관리로서 감사증적이 유지되고 변경통제절차 등이 적절히 적용되는지 여부와 그리고 개발 유지단계가 개발기준에 부합하는지 여부를 체크한다. 즉 업무프로그램의 변조, 파괴 등을 방지하고 프로그램 보호와 변경이력관리를 위한 것이다. 데이터베이스관리통제는 데이터에 대한 갱신 및 접근보안을 유지하고 데이터베이스의 논리적 및 물리적 구조에 대한 보안유지를 의미한다. 파일반출관리, 파일매체관리, 데이터세트관리, 자원사용자관리 등의 내용이 포함된다.

사례연구결과는 4장 및 5장에 제시하였다. 4.1절 및 5.1 절에서는 시스템 개요를 설명하였고 4.2 절과 5.2절에서 각 은행시스템의 주요보안위험을 면담(4.2절 및 5.2절) 및 통계분석(5.2절)을 통하여 파

악한 결과를 설명하였고 각 보안위험에 대한 통제 방안을 제시하였다.

4. A은행 사례적용

4.1 A은행 정보시스템 개요

본 연구의 첫번째 사례는 클라이언트-서버구조에 의한 다운사이징을 실현한 A은행 금융 전산망이다. A은행 금융전산망은 클라이언트-서버 시스템 구축 후에 사용자 컴퓨팅, GUI, 부하분산이 가능해졌고 중소형 컴퓨터의 조합에 의한 시스템구축으로 전산투자비용을 획기적으로 절감했으며 정보처리의 생산성을 제고시켰다. 클라이언트-서버 구조를 통한 분산업무처리는 정보계 시스템에서 이루어지고 있다. 유닉스 운영체제를 사용해서 금융소프트웨어의 하드웨어의 종속성을 피하는 개방형 시스템이다. A은행은 종합정보시스템화로 고객서비스체계의 일원화를 추구하고 전자은행화로 금융환경 변화에 대응하며 멀티미디어화를 추진하고 있다.

A은행의 정보계 시스템은 소프트웨어 구조를 볼 때 서버에서 데이터베이스관리 및 응용시스템 일부(데이터 로직, 업무로직)가 실행되고 클라이언트에서 나머지 응용시스템과 업무로직이 실행되는 로직분할 구조로 구축되어 있다.

4.2 분석 결과

다운사이징(Downsizing)의 실현으로 A은행은 클라이언트-서버 시스템을 이용한 분산처리를 구현했으나 통제위험도 증가해서 시스템 및 네트워크의 안정화가 필요하게 되었다. 시스템위험은 다음과 같이 세 가지로 분류되어 제시될 수 있다. 응용시스템관리 및 데이터베이스관리에 관련된 위험(혹은 통제부재상황) 을 각각 먼저 제시하고 두 가지 통제에 모두 관련된 위험을 제시하면 다음과 같다.

4.2.1 응용시스템관리에 관련된 위험

1) 응용프로그램의 임의변경 위험

실행프로그램 변경승인절차는 있으나, 변경내용의 정확성에 대하여 확인하는 방법이 정립되어 있지 않다. 또한 야간작업시 프로그램 변경이 긴급히 필요한 경우에, 선조치 후보고하도록 되어 있고 미보고시 확인이 어렵다. 그리고 업무분야별로 공통의 계정을 사용한다. 개발 환경에서 원시프로그램이 개발 담당자들에게 무방비 상태로 노출되어 있다.

결국 프로그램에 대한 불법적인 변경의 위험이 있고 프로그램의 불법적인 변경에 대한 발견이 이루어져도 변경자에 대한 파악이 어렵다. 또한 원시프로그램의 외부유출의 위험이 있다.

4.2.2 데이터베이스관리에 관련된 위험

1) 대차불일치 문제

장부상의 대차가 불일치하는 경우가 자주 발생하여 원인규명을 통한 원장변경 처리를 하고 있다. 문제해결을 위해 개선팀이 구성되어 문제거래에 관련된 응용시스템을 재개발하고 본부서버에서는 특정 시간대에 집중적으로 이루어지는 거래에 대한 분산처리 등을 실시하고 있다.

관련된 문제점으로는 1) 원장마감을 위하여 대차불일치의 원인파악에 시간소모가 크다는 것과 2) 응용프로그램에 의한 처리가 아닌 DBA(Database Administrator)에 의하여 원장변경이 수행됨으로써 금융사고 가능성이 있고 3) 고객 불만으로 인한 은행의 신용도가 실추할 위험이 있다.

2) DB원장의 임의변경위험

비승인된 원장변경이 있어도 사후발견이 어렵고 원장변경관련 문제일지가 작성되지 않고 있으며 DB담당자간에 공동의 유닉스(UNIX) 계정을 사용하고 원장변경을 SQL을 사용하여 수행하고 있다. 원장변경에 대한 통제에 관련된 오라클(Oracle)의 감사(Audit)기능이 사용되지 않고 있다(Audit Log가 없음). 또한 임의 원장변경 여부를 파악하기 어렵고 원장변경을 알았어도 누가했는지를 파악하기

어려우므로 임의의 구좌와 거래처리를 발생시킬 수 있는 위험이 있다. 그리고 비정상거래가 발생시 발견이 어렵고 담당자 결백이 증명되기 어렵다.

3) 백업의 불완전

백업 테이프의 복구가 부정기적으로 실시되고 있고 백업 담당자가 시스템별로 분산되어 있다. 일주일 단위로 지역센터의 백업테이프가 교체되어 보관되고 백업 테이프를 시스템 하드웨어와 같이 보관된다. 장시간의 백업시간이 소요 (1회 4시간 정도)되고 백업테이프는 1부만 제작된다. 백업시간이 장시간 소요됨에 따라 백업테이프가 1부만 제작되고 원본상실시 데이터 복구비용이 크다. 그리고 지역센터의 백업데이터의 교체보관 주기인 일주일 이내에 재난이 발생하는 경우에는 데이터 복구비용이 클 위험이 있다.

4.2.3 응용시스템 및 데이터베이스관리에 전반적으로 관련된 위험

1) 영업점 및 시스템 패스워드관리 취약

정보지원팀 내의 클라이언트 통합관리자는 영업점에서 사용되는 패스워드가 기록된 파일을 조회하거나 출력할 수 있다. 또한 패스워드가 4자리밖에 안되고 많은 창구담당자의 패스워드 작성이 허술하며 패스워드 노출가능성이 높다. 그리고 로그-온 시에 패스워드가 틀리더라도 실질적으로 계속 시도해 볼 수 있다. 결국 창구담당업무와 관련된 컴퓨터 범죄발생 위험이 크고(예, 다른 사람의 패스워드를 사용한 무단현금인출) 컴퓨터 범죄발생시 범죄자 파악이 어렵다. 또한 담당자 무협의가 보호되지 않을 위험(결백증명이 어려움)이 있다.

시스템 패스워드의 경우에 하나의 계정이 여러 사용자에 의해 공용되고 있고 분석결과 11개의 계정 중 2개는 패스워드가 없고 1 계정의 패스워드는 쉽게 추정이 된다는 결과를 얻었다. 결국 프로그램과 데이터 파일을 임의로 조회하거나 수정, 변경할 수 있는 위험이 있으며 사고발생시 추적이 곤란하다. 또한 A은행의 전산시스템은 기본적으로 유닉스(UNIX)를 사용하는 개방시스템으로서 본부-지점간 어느 네트워크 회선에서도 다이얼-업 침투의 위험이 있다.

2) 비상계획의 취약

각종 재난에 대한 위험분석이 이루어져 있지 않다. 그리고 비상사태시 처리절차, 복구순위, 복구절차 등을 제시하는 비상계획이 없다 또한 비상연락망에 은행전체의 주요 임원이나 직원에 대한 우선순위경로가 명시되어 있지 않고 있다. 재해복구계획과 이에 따른 재해대책 테스트가 없다.

결국 재해발생시 효과적 대응이 어렵고 그에 따라 은행에 큰 손실이 초래될 수 있다. 또한 재난시 복구시간이 장기화될 수 있고 재해복구비용이 증가할 수 있다.

3) 시스템 운영상 미비점

시스템 로그에 대한 정기적 검토가 이루어지지 않고 문제발생시에만 검토가 이루어지고 있다. 그리고 프로그램 및 데이터파일의 기밀성에 따른 분류가 안되고 있다. 마지막으로 시스템 소프트웨어의 변경통제가 미흡하고 출력물 관리통제가 미약하여 출력 데이터가 외부로 유출될 가능성이 있다. 결국 미보고된 문제의 발생가능성이 있고 프로그램 및 데이터의 외부유출 가능성이 높다.

일부 시스템에서는 로그인후 업무자의 장시간 이석시 시스템이 자동로그오프되는 기능이 부재한 상태이고 명문화된 업무순서 수립(Job Scheduling) 절차가 미비하다. 시스템운영자의 장시간 이석시 타인에 의한 악용, 오용 가능성이 존재하고 업무순서 수립을 수작업으로 처리함에 따라 전산요원에게 과다하게 업무가 주어지거나 업무가 잘못된 작업순서에 의해 이루어지고 업무처리가 누락이 될 위험이 있다. 잘못된 복구절차 수행에 따른 중대한 시스템 오류발생 위험이 있고 담당자의 과실 또는 고의적 시스템 변경 행위로부터 시스템의 안정성에 대한 보장이 어렵다.

〈표 2〉 A 은행 정보시스템의 주요 위험과 구체적인 통제방안

위험분류	A 은행 정보시스템의 주요 위험	구체적인 통제방안
응용시스템관리에 관련된 위험	프로그램의 임의변경 위 험	● 보안소프트웨어(Security Tool Kit) 으로 수시점검 ● 개별담당자별 계정 지정 ● 프로그램 라이브러리 관리 소프트웨어를 사용 ● 소스프로그램 버전관리 절차수립
데이터베이스 관리에 관련된 위 험	대차불일치문제	● 기존의 발생건 형태별로 오류상태를 문서화 ● 오류유형별로 지속적인 통제 ● 1일 보고체계를 구축(전산감사팀에 의한) ● 전담팀(2인 이상)을 구성하고 외부자문용역필요(매월 전산감사팀에 보고)
	DB원장의임의변경 위 험	● SQL 사용 통제절차의 수립 ● DB담당자별 고유의 유닉스(UNIX) 계정의 사용 ● DB원장관련 문제일지의 작성 ● 원장변경을 프로그램으로 수행토록(감사증적 확보) 개발
	백업의 불완전	● 백업 자동화 등의 방안 강구 ● 백업 테이프의 정기적 점검을 공식절차화함 ● 지역센타별 소산영업점을 선정하여 매일 소산실시 ● 오프사이트(Off-Site) 백업 필요 (외부 소산) ● 전산매체의 정기적 실사 ● 테이프 관리소프트웨어(Tape Library Software)도입 ● 전산매체 표준레이블 제작 사용 ● 백업 테이프와 기계실외에 다른 장소에 보관
응용시스템 및 데이터베이스 관리에전반적으로 관련된 위 험	영업점 및 시스템 패스워드관리 취약	< 영업점 패스워드 강화방안 > ● 패스워드 중요성에 대한 교육강화 ● 패스워드 정보는 조회, 출력이 불가능하도록 조치(암호화) ● 최소 6자리 이상 숫자와 문자를 혼용토록 함 ● 강제적으로 변경이 되도록 함(패스워드 변경 소프트웨어 사용) ● 로그-온 시에 패스워드를 무제한 시도하지 못하도록 실질적인 통제강화 < 시스템 패스워드 강화방안 > ● 공용 Account사용중지(사용별로 다른 Account지정) ● 시스템패스워드 Account 관리지침 정립 ● 주기적 시스템 Tool Kit점검 ● 쉽게 추정이 가능한 Account패스워드는 변경조치 ● 접근통제소프트웨어 설치운용 ● 주기적 패스워드변경이 가능토록 함(패스워드변경 소프트웨어 사용) ● 패스워드, 무제한 시도 통제 ● SAS 클라이언트에도 패스워드 사용
	비상계획의 취약	● 비상계획을 체계적 방법론을 통하여 수립하여야 함 ● 재해복구관련 소프트웨어 활용 ● 정기적, 부정기적으로 재해대책 테스트를 수행하여야 함 ● 정보지원팀 내에 담당자 임명 ● 컴퓨터 재난시의 비상연락망의 우선순위 설정
	시스템 운영상 미비점	● 파일의 중요도에 따른 기밀성 부여 ● 시스템 생성로그의 파악 및 분석 ● 자동 로그오프기능 구현 ● 업무순서 수립과 승인절차 수립 ● 장애복구절차 표준화 ● 출력물 통제절차 정립 ● 시스템 소프트웨어 표준적용 및 변경절차 수립
	물리적 보안위험	● 화재방지 보안(연기감지기, 자동소화설비, 항온 항습기능 추가) ● 출입통제구역 물리적 통제 보완 (기계실 최종출입문 등 카드키 설비) ● 통제구역 출입관리제도 (카드키 발급, 회수절차 등)

4) 물리적 보안위험

기계실에 연기감지기가 없고 기계실 및 운영실에 자동소화설비(가스)가 필요하다. 기계실에 최종 출입문 카드키가 없고 운영실의 경우에 출입문에 감시용 카메라가 설치될 필요가 있다. 이러한 미비점으로 인하여 화재발생시 물리적 피해가능성이 있고 기계실에 외부인 출입가능성이 있다. 운영실의 경우 카드키를 소지하지 않은 무단 출입에 대한 적발이 어렵다.

<표 2>에서 제시한 바와 같이 이러한 여덟 가지 위험은 효과적인 응용시스템관리 및 데이터베이스관리가 필요함을 직접 및 간접적으로 나타낸다. 프로그램의 임의변경위험, 대차불일치문제, DB원장의 임의변경위험, 백업의 불완전은 직접적으로 각각 응용시스템관리통제 및 데이터베이스관리의 필요성을 나타낸다.

이러한 문제점에서 가장 심각한 것은 대차의 불일치 문제와 DB원장의 무작위 변경문제이다. 이 두 가지 문제가 다른 문제보다도 중요한 이유는 은행업무에서 가장 민감한 원장데이터관리에 관한 위험이기 때문이고 A은행의 기본적 운영체제인 유닉스(UNIX) 및 오라클(Oracle) 시스템의 취약점을 반영하기 때문이다.

결국 A은행의 정보계시스템은 사용자 및 운영자들이 각 클라이언트에서 응용시스템을 개발하여 업무를 처리하므로 응용시스템 및 데이터에 대한 통제가 일관되게 이루어지지 못하게 되는 가능성이 커졌다. 이는 결국 시스템의 무결성 및 정확성에 문제를 야기시키게 되었다. A 은행 정보계시스템의 주요 위험과 이에 대한 구체적인 통제방안을 제시하면 <표 2>와 같다. <표 2>에서 제시된 바와 같이 통제방안 들의 대부분은 전반적인 관리통제에 해당되는 비상계획 및 물리적 보안의 일부통제를 제외하고는 응용시스템 및 데이터관리통제에 관련되어 이 두 가지 부분에 대한 통제가 중요하다는 것을 보여준다.

정보시스템 위험에 영향을 미치는 요소로는 처리의 복잡성, 거래량, 거래의 민감성, 응용시스템 오류의 영향 등이 포함된다(Weber, 1999). 처리의 복잡도 측면에서 보면 서버 및 클라이언트 쪽에 응용시스템이 분산되어 있는 경우가 처리복잡도가 크다. 로직분할 구조는 클라이언트쪽에 응용프로그램이 분산되거나 응용프로그램이 서버와 클라이언트에 나뉘어진 구조이므로 응용프로그램의 관리가 어려워진다. 프로그램 관리의 복잡성은 관리의 어려움을 증가시키고 변경통제나 유지통제와 같은 응용시스템관리통제에 대한 중요성을 증가시킨다. 클라이언트쪽에 응용프로그램에 의한 처리가 많으면 프로그램 오류발생 등으로 인한 파급 위험의 정도는 커지므로 이에 대한 관리통제가 중요해질 것이다.

응용시스템 로직이 클라이언트에서 이루어지면 분산화 되어 처리되는 데이터에 대한 통제가 필요하다. 분산화된 컴퓨터시스템에서는 메인프레임을 사용한 중앙집중식 시스템보다 데이터베이스관리 위험이 높다(Boockholdt, 1987). 클라이언트에 있는 소형컴퓨터기종에 있는 데이터들은 비숙련된 사용자들에 의해 관리되어질 수 있어 데이터의 보안위험이 높다. 대부분 데이터파일이 서버에 없고 클라이언트에 있으므로 데이터통제의 중요성이 커진다.

5. B은행 사례적용

5.1 B은행 정보시스템 개요

B은행의 계정계시스템은 중앙집중식 시스템으로 구성되어 있으나 정보계시스템은 클라이언트-서버 구조로 되어 있다. 정보계업무는 최종사용자 컴퓨팅(End User Computing)을 위해 클라이언트-서버 시스템을 구축하여 투자정보업무는 이를 통해서 이루어지고 있다. 클라이언트-서버 아키텍처는 응용프로그램이 대부분 클라이언트에 상주하고 데이터는 서버에 위치하는 대표적인 원격데이터 구조를 하고 있다.

지역서버가 일곱대로 전국 4개 지역에 설치되어

있다. B 은행 전 직원의 수(이용자 수)는 약 2,000명이고, 정보 지원 팀(운영 요원)의 수는 약 90~100명 정도이다. 각 서버는 7개 지역 센터에서 서로 원격백업저장소(Off-site Backup Storage) 역할을 수행하고 있다. 그리고, 각 서버의 소프트웨어는 거의 동일한 것을 사용하여 분산처리를 수행하고 있으며 집계, 통계 분석은 그 중 하나인 중앙서버에서 수행한다.

5.2 분석 결과

본 사례에서는 계정계시스템과 최종사용자 컴퓨팅 구현을 위해 클라이언트-서버 시스템을 구축한 정보계 시스템의 통제수준을 비교하여 정보계시스템 즉 클라이언트-서버 시스템에서 필요한 통제를 분석하였다. B은행에 대한 분석은 설문조사를 통한 면담분석과 통계적 분석방법론을 사용하였다. A 은행사례연구에서 사용한 면담방식 외에 통계적 분석을 추가적으로 사용함으로써 보다 일반화된 결론을 도출하는데 기여할 수 있을 것이라고 판단된다. 설문문항은 응용시스템관리, 데이터베이스관리의 두 가지 부문으로 나뉘어 30여 문항으로 구성되었다. 전산부직원 130여명에 대한 보안관련 설문조사를 실시하여 99부가 회수되었다 (응답율 = 76.15%). 이중 9개는 응답내용이 충분하지 못하여 총 90개가 분석에 사용되어졌다. 표본의 특성 중 응답자의 근무 년수가 <표 3>에 제시되어 있다.

설문서 데이터에 대한 통계분석 결과가 표4에 제시되었다. 정보계시스템과 계정계시스템 및 전체시스템 (정보계와 계정계의 평균)의 응용시스템 관리통제, 데이터베이스관리통제의 인지된 수준을 t검정 (Paired t Test) 으로 비교하였다. 대부분의 경우에서 95%신뢰구간에서 차이가 유의 한 것으로 결과가 제시되었다. <표 4>에 제시된 바와 같이 클라이언트-서버 시스템으로 구축된 정보계시스템이 메인프레임 방식으로 되어 있는 계정계 시시스템 보다 각 통제부문 응용시스템관리통제, 데이터베이스관리통제 의 통제의 수준이 유의적으로

〈표 3〉 B은행 응답자의 근무 년수

현업무 근무년수	빈도수	응답자 비율 (%)
0	13	14.4
1	34	37.8
2	12	13.3
3	13	14.4
4	9	10
5	8	8.9
8	1	1.1
합 계	90	100

낮다는 결과를 나타낸다. 또한 통제의 유용성도 정보계시스템의 통제의 유용성이 계정계보다 유의적으로 낮다. 이는 클라이언트-서버 시스템으로 구축된 정보계시스템에 대한 응용시스템관리 및 데이터베이스관리통제의 만족도 및 효과성이 계정계에 비해서 낮다는 것을 나타낸다. 결국 클라이언트-서버 시스템의 통제는 수준 뿐만 아니라 질적으로도 향상될 필요가 있음을 지적한다. 클라이언트-서버 시스템은 중앙집중식 시스템에 비해서 전반적으로 사용자 만족도가 높지만 시스템 통제에 대한 유용성 및 만족도는 낮다는 사실을 나타낸다.

정보계시스템은 최종사용자 컴퓨팅 구현을 위해 클라이언트-서버구조를 구현하고 있으며 조회, 분석, 보고서작성, 계산 등의 응용로직의 대부분이 클라이언트에 상주하고 데이터는 서버에 위치하는 원격데이터 구조이다. 면담조사결과 정보계시스템에서 응용시스템관리 및 데이터베이스관리가 중요한 반면에 이부문의 통제정도가 상대적으로 낮다고 지적되어 이 두 가지 통제부문이 중요함을 나타내었다. 응용시스템 관리통제가 낮은 것은 최종사용자 부서에서 자체 개발한 프로그램은 클라이언트에 상주하고 따라서 전산부서 경영진에서 체계적으로 관리하는 것이 아직 어렵기 때문이다. 데이터베이스 통제 정도가 낮은 것은 상대적으로 원장데이터에 대한 통제의 기대수준이 큰 이유도 있지만 분산되어 있는 클라이언트에 존재하는 프로그램에서의 처리를 위해서 데이터가 서버를 벗어

〈표 4〉 B은행 정보계 및 계정계 보안시스템 분석(Paired t TEST)

변수간의 평균차이 (Paired t TEST)	평 균	t 값	2 tailed significance	차이의 유의도 여부(95%신뢰구간)
정보계 응용시스템관리통제수준 계정계 응용시스템관리통제수준	3.89 4.90	7.37	0.000	유의
정보계 데이터베이스관리통제수준 계정계 데이터베이스관리통제수준	3.68 4.66	6.12	0.000	유의
정보계 응용시스템관리통제수준 전반적인 보안수준	3.88 4.27	3.25	0.002	유의
정보계 데이터베이스관리통제수준 전반적인 보안수준	3.61 4.27	4.08	0.000	유의
정보계 통제유용성 (효과성) 계정계 통제유용성 (효과성)	4.10 4.44	4.58	0.000	유의

나서 이동되어야 하므로 데이터 오류 및 보안위험이 크기 때문이다.

B은행 네트워크가 현재 패쇄환경 이어서 보안위험이 적고 현재 사용자 부서에서 업무처리를 위해 데이터를 서버에서 가져오는 것이 아니라 자체적으로 개인 DB를 구축해서 자체 DB로부터 데이터를 사용함으로써 네트워크상의 빈번한 데이터 이동은 없다. 그러나 향후에 최종사용자에 의한 업무처리 및 프로그램 개발이 활성화될 경우에 데이터 및 프로그램 변경업무가 많아질 것이고 따라서 데이터베이스 및 응용시스템관리 부문에 대한 통제가 점점 중요해질 것이다. 실제로 정보계의 최종사용자 컴퓨팅관련 클라이언트-서버 아키텍처의 담당 관리자를 면담한 결과, 앞으로 시스템의 외부환경으로 접속이 점점 넓어지는 개방환경으로 나아가게 되면 해커나 외부침입자의 다이얼-모뎀 등을 통한 침투가 광범위해 질 수 있다고 지적됐다.

정보계시스템의 전반적인 보안수준(4, 10)은 실제 금액의 처리를 대상으로 하는 계정계시스템 보다(4.55) 낮게 평가되었다. 아직 최종사용자컴퓨팅과 클라이언트-서버구조에 대한 통제가 정착되지 못한 이유도 있지만 개방시스템이 가지는 보안상의 취약점을 반영하는 것으로 해석되었다.

B 은행의 클라이언트-서버 시스템의 응용시스템관리 위험과 데이터관리위험 그리고 이에 대한 대응방안을 살펴보면 다음과 같다.

5.2.1 응용시스템 관리위험 및 통제방안분석

클라이언트-서버 시스템에 대한 개발기술의 부족과 전문가의 부족이 시스템 설계부분에 결함을 초래한다. 또한 클라이언트-서버 시스템에서 응용시스템 개발은 신속한 개발 절차(Rapid Application Development ; RAD)의 절차를 거쳐 신속하게 개발되어지는 것이 보통이다. 공식적이고 구조화된 개발절차에 따라 시스템 평가를 거쳐 개발되어지는 것이 아니라 시스템의 신속한 가동을 위해 RAD를 따르는 것이다. 또한 워크스테이션이나 개인용 컴퓨터에 대한 해커들의 침입가능성으로 프로그램 무결성의 위험이 커지고 있다.

프로그램 사서함 통제를 통한 사용자 응용시스템의 객체 코드(object code)에 대한 변경통제가 이루어지고 프로그램에 대한 무결성 통제가 이루어져야 한다. 가장 흔한 통제방식으로는 패스워드에 의한 프로그램 접근통제방식이 있다(Lyons, 1994 ; Symonds, 1994).

프로그램개발관리는 보안조직관리, 보안인식교육 등의 관리적 측면의 노력이 필요하다(Davidson, 1994 ; Fried, 1993). 분산화된 환경의 보안은 기술적인 문제가 중요하기는 하지만 이를 뒷받침하는 조직적 관리적 직무체계가 없으면 어렵다. 분산화된 정보시스템 환경에서 정보보안에 관한 조직업무에서는 네트워크의 가용성을 체크하고 기술적 보안서비스 지원을 하는 네트워크 제공자, 기밀성

을 체크하는 응용시스템 소유자(application owner)와 전산처리자원 및 시스템 운영자 등의 업무를 명시하도록 하여야 한다(Fried, 1993). 시스템 운영자(system administrator)로 하여금 시스템에 대한 전권을 부여하도록 하는 경우에는 업무분장이나 시스템 운영에 대한 통제가 요구된다. 즉 이러한 시스템 운영자나 동일한 권한을 가진 사용자에 대한 확인 및 업무 적정성 감사가 요구된다. 각종 서버에 대한 합계통제나 감사루틴 및 위반 로그를 유지하고 사용하는 것도 효과적일 것이다.

분산된 클라이언트에서 개발된 시스템은 중앙집중식 시스템에서 개발된 응용시스템보다 시험단계를 철저히 거치지 않은 경우가 많기 때문에(Boockholdt, 1987) 응용시스템 오류의 가능성이 높아지므로 클라이언트에 분산되어 있는 응용시스템 통제의 중요성이 크다. 응용시스템을 관리하고 통제하는데 우수한 인력과 기술이 필요하다.

5.2.2 데이터관리위험 및 통제방안분석

사용자가 파일서버에 로그인 할 때 입력된 패스워드가 중간에 노출되는 위험이 있을 수 있다. 네트워크에서 전송은 정확히 목적지에 전송이 되지 않거나 전송 중에 도청 되거나 손상을 받을 위험이 존재한다. 클라이언트와 서버를 잇는 연결 네트워크 중간에 대한 침투위험은 대체적으로 데이터량이 빈번히 오가거나 네트워크선이 많이 존재하는 경우에 증가한다고 볼 수 있다(Runge, 1994 ; Ryans, 1992 ; Teplitzky, 1994). 이러한 네트워크에 대한 불법적인 접근은 네트워크에 전송되는 데이터통신에 대한 특성(예 : 프로토콜, 전송량, 빈도) 을 분석하고 분석된 특성을 이용하여 합법적인 사용자임을 가장하는 경우가 있다. 원격데이터 구조는 네트워크 부하가 상대적으로 다른 구조에 비해서 높다. 네트워크 부하가 높으면 네트워크에 위험이 발생할 경우에 손실정도가 커진다고 볼 수 있다. 네트워크에 전송되는 데이터양이 많으면 데이터에 대한 보안위험의 가능성이 커지는 것으로 생각할 수 있다. 클라이언트에 의존율이 높으므로 분산화 되어 있는 클라이언트에 대한 통제가 필요하다.

DBMS를 통해서 운영체제를 거치지 않고 직접 응용시스템 데이터에 접근이 가능한 경우에 DBMS의 보안기능이 중요시된다. 조직의 데이터를 기밀도에 의해 분류하고 이러한 데이터의 기밀도는 클라이언트와 서버에 적용되어야 한다. 클라이언트-서버환경에서는 다이얼-업(dial-up)서비스가 많이 제공되어지는데 이에 대한 패스워드나 사용자 ID를 통한 통제가 요구된다. 스마트 카드(smart card)나 콜-백(call-back) 시스템이 불법적인 다이얼-업의 위험을 줄일 수 있다. 또 다른 방법은 ACL(Access Control List)를 정의한 게이트웨이(gateway)를 통해 네트워크에 접근을 시도하려는 사용자에 대한 인증통제방법을 사용하는 것이다.

원격데이터 구조는 데이터가 서버와 클라이언트 사이에서 오고 가는 형태로서 서버에서 벗어나 있게 된다. 데이터 보안을 위해서 클라이언트에서 사용자가 입력 및 접근하는 것을 제한하기 위한 화면설계나 프로그램상의 통제가 중요하다. 이것은 클라이언트측의 업무처리양에 비례하여 중요해진다. 클라이언트쪽에 거래양이나 데이터량이 많아지면서 데이터의 무결성 및 일관성을 위한 물리적 및 논리적 통제가 중요해진다.

6. 결 론

본 연구에서는 클라이언트-서버아키텍처의 특성을 분석해서 그러한 특성때문에 생길 수 있는 위험을 분석하고 그러한 위험에 대한 통제방안을 제시하였다. 클라이언트 서버 아키텍처인 로직분할 및 원격 데이터구조는 응용업무프로그램을 클라이언트에 분산시켜서 유지해야 하므로 응용시스템 관리통제가 중요하다. 또한 클라이언트에서 데이터가 분산되어 처리되므로 데이터베이스관리통제가 중요하다. 즉 클라이언트에 분산되어 존재하는 프로그램에서 처리되는 데이터는 서버로부터

전송되어서 클라이언트에 존재해야 하므로 데이터 관리통제가 중요하다.

클라이언트-서버 시스템을 구축한 A은행과 B은행의 사례분석을 통해 클라이언트-서버 시스템에 대한 위험 및 통제를 분석하고 대응방안을 제시하였다. A은행의 경우에는 정보계시스템의 경우 로직분할 구조로 되어 있고 보안실사의 결과 원장데이터 무결성, 클라이언트 패스워드 관리, 응용시스템 유지 관리 등의 문제점이 크게 나타나서 응용시스템 및 데이터베이스관리통제 방안이 필요함을 나타냈다. B은행의 정보계시스템인 경우에는 원격데이터 구조로 구축되었는데 중앙집중식 메인프레임시스템으로 구성되어 있는 계정계시스템에 비하여 응용시스템 및 데이터베이스 관리통제수준은 낮았고 통제의 만족도 및 효과성도 낮게 평가되어 응용시스템 및 데이터베이스 관리통제가 중요함을 나타냈다. 클라이언트에서의 보안을 강화하기 위해서 결국 클라이언트에서의 패스워드 관리 및 클라이언트 PC에 대한 보안관리를 효과적으로 해야 한다. 그리고 이러한 보안정책을 수립하기 위해 필수적으로 통제에 대한 문서화를 추진하여야 한다.

사례에서 제시된 클라이언트-서버 시스템은 조직의 분산화를 유도하고 최종 사용자 및 운영자에게 보다 많은 전산 업무능력과 통제권한을 부여하게 만들었다. 업무의 생산성 및 처리의 유연성을 높이기 위하여 다양한 플랫폼, 운영체제, 프로토콜 및 응용시스템으로 구성되었으나 데이터 및 응용프로그램에 불법적으로 접근할 수 있는 경로와 가능성이 높아졌다. 보다 많은 권한을 부여 받은 최종 사용자들의 시스템 사용능력 및 통제에 대한 중요성 인식은 아직 부족하여 보안위험이 증가했다. 예를 들면 프로그램 개발 및 변경이 적절한 인가 절차가 없이 이루어지고 데이터의 입력 및 수정이 체계적인 승인절차 없이 이루어지는 경우가 있었다. 또한 최종사용자 및 운영자들은 아직 패스워드, ID, 로그온 절차, 그리고 접근권한의 관리 중요성을 인식하지 못하고 있다. 아직은 이러한 위험이 구체적인 손실이나 재해를 가져오지는 않았지만 향후에 클라이언트-서버 시스템이 조직에서 차지하는 역할이 커지고 시스템의 오용을 통해 얻을 수 있는 이익이 커진다면 분산된 응용시스템이나 데이터베이스에 대한 적절하지 못한 통제는 조직에 커다란 재해를 가져올 수 있다. 클라이언트-서버 시스템의 통제방안은 여러 사회적인 요인(컴퓨터 범죄의 증가, 해커로 인한 국가적인 피해의 증대 등)으로 필요성이 높아지게 될 것이다. 본 사례연구는 응용시스템 및 데이터베이스관리측면에서 보안위험을 제시하고 적절한 통제방안을 수립하는데 기여했다.

미래의 연구에서는 클라이언트-서버 시스템통제에 대하여 기술적인 면 외에도 조직적, 문화적인 고려해야 한다. 보안시스템 구축에 있어서 중요한 이슈중의 하나는 조직차원의 문제로 새로운 통제시스템에 대한 이해부족 및 적응에 대한 거부감 그리고 조직구성원의 오해 그리고 반발 등이다. 이러한 문제의 해결을 위해서 사용자 및 실무자에 대한 교육 및 홍보활동을 통해 통제의 중요성을 부각시켜야 할 것이다. 그리고 미래의 연구는 실제로 다수의 기업 및 다양한 클라이언트-서버 아키텍처를 대상으로 이루어질 필요가 있다. 예를 들면 클라이언트 및 서버 시스템에서 응용시스템, 데이터베이스에 통제의 중요성은 클라이언트-서버 시스템의 중요한 수행프로그램이 클라이언트 혹은 서버에 분리된 정도에 따라 달라질 수 있을 것이다. 즉 클라이언트에 응용시스템, 데이터베이스관리시스템, 프레젠테이션서비스 등이 집중되어 있으면 분산된 사용자 및 프로그램, 데이터에 대한 통제가 더 필요하다. 반대로 서버에 클라이언트-서버 시스템의 중요한 수행프로그램이 집중된 경우에 상대적으로 통제관리가 수월해지지만 집중된 데이터 및 프로그램에 대하여 비상계획 및 복구계획의 중요성이 커질 것이다. 또한 클라이언트-서버 시스템 통제에 대한 연구가 많은 규모의 샘플을 이용한 통계적 실증분석을 하는 형태로 이루어진다면 보다 일반화된 결론을 도출을 하는데 기여할 수 있을 것이다.

클라이언트-서버 시스템은 데이터베이스, 네트워크, 응용시스템 등이 복합적으로 구성된 시스템이다. 이러한 시스템의 통제를 평가하기 위해 응용프로그램 및 데이터베이스관리 외에 다른 통제요소를(예를 들면 네트워크나 단말기 관리통제) 추가하거나 다른 통제분류차원을 사용할 필요가 있다. 다른 통제분류차원의 예로서는 관리 및 응용통제가 있다. 관리통제는 전반적인 통제환경과 관련이 있고 응용통제는 각각 클라이언트-서버 시스템의 구조와 특정 응용프로그램의 통제와 직접 연결된다. 따라서 관리통제는 클라이언트-서버 시스템을 도입한 기업의 전산 업무처리에 공통으로 적용되는 통제로서 응용통제를 수립하기 위한 기초가 된다고 할 수 있다. 관리 및 응용통제는 서로 독립적으로 평가되어질 수 있으나 서로 연관관계를 가진다. 클라이언트-서버 시스템의 이러한 계층화된 통제모형은 여러 가지 다른 클라이언트-서버 구조를 평가하거나 벤치마킹 하는데 쓰일 수 있다.

참 고 문 헌

[1] Anandarajan, M. and B. Arinze, "Matching Client/Server Processing Architectures with Information Processing Requirements : A Contingency Study," *Information & Management*, Vol.34(1998), pp.265-274

[2] Atre, S., "Twelve Steps to Successful Client-server," *DBMS*, May(1994). pp.70-76.

[3] Boockholdt J.L., "Security and Integrity Controls for Microcomputers : A Summary Analysis," *Information & Management*, Vol. 13(1987), pp.33-41.

[4] Bruce, M.C., "PC Security Criteria A to Z," *IS Audit & Control Journal*, Vol.5(1995), pp. 27-32.

[5] Cairo, L. and A. Friedberg, "Security in Client/Server Authentication Issues," *IS Audit & Control Journal*, Vol.4(1995), pp.48-53.

[6] Chengalur-Smith, I., P. Duchessi, "The Initiation and Adoption of Client-Server Technology in Organizations," *Information & Management*, Vol.35(1999), pp.77-88.

[7] Davidson, A.M., "Security in an Oracle Data Base Environment," *Information Systems Security*, Winter(1994), pp.59-68.

[8] Duncan, H. and D. Sidwell, "Distributed Database Security," *Computers & Security*, Vol. 13(1994), pp.547-557.

[9] Dykman A.C. and. C.K. Davis *Control Objectives*, The EDP Auditors Foundation, Inc., 1992.

[10] Edelstein, H., "Unraveling Client Server Architecture," *DBMS*, May(1994), pp 34-42.

[11] Fried, L., "Distributed Information Security," *Information Systems Management*, Summer (1993), pp.56-65.

[12] Gartner Group, "Best C/S Practices," *Datamation*, (1994), pp.24-32.

[13] Hale, R., "End-User Computing Security Guidelines," *IS Security*, Winter(1996), pp.49-64.

[14] Hancock M.W., "Dial-up Modem Protection Scheme," *Information Systems Security*, Winter(1994), pp.42-51.

[15] Held, G., "Cooperative Processing : the Key to Efficient Client-server Operation," *Information Strategy : The Executive's Journal*, Summer. (1994), pp.29-33.

[16] Lainhart, W.J. and M. Donahue *Computerized Information Systems (CIS) Audit Manual*, The EDP Auditors Foundation, Inc., 1992.

[17] Lyons, W.F., "An Introduction to Client/server Security," *DATAPRO Information Security Service*, September (1994), pp.101-105.

[18] Moeller, R.R., *Computer Audit, Control, and Security*, John Wiley & Sons, 1989.

[19] Mollini, E.J., "The New Frontier : Client-Server and Agent Technology," *Information Systems Security*, Fall(1994), pp.64-71.

[20] Runge, L., "Security and Data Integrity in a Client-server Environment," *Information Systems Security*, Spring (1994), pp.45-56.

[21] Ryan, S.D., B. Bordoloi "Evaluating Security Threats in Mainframe and Client/Server Environments," *Information & Management*, Vol. 32(1997), pp.137-146.

[22] Ryan, W.H., "Issues in Client-server Development," *Information Systems Management*, Fall(1992), pp.48-50.

[23] Symonds, M.L., "Security in Distributed and Client/server Systems-A Management View," *Computers $ Security*, Vol.13(1994), pp.473-480.

[24] Teplitzky, P., "Solutions for Client-server Security," *Information Systems Security*, Winter(1994), pp.21-27.

[25] Weber, R., *Information Systems Control and Audit*, Prentice Hall Inc., Upper Saddle River, New Jersey, 1999.

[26] Wood, C.C., "Fifty Ways to Secure Dial-up Connection," *Computers & Security*, Vol. 13 (1994), pp. 209-215.

[27] Yin, R.K., *Case Study Research*, SAGE Publications, 1993.

〔부 록〕

1. 통 제

각 통제항목에 대해 현재 통제수준과 통제의 중요도를 리커르트 7점 척도로 측정하였음.

1) 응용시스템 관리통제

(1) 프로그램의 수정이 사용자의 문서화된 요구와 해당팀장, 시스템운영팀장, 정보기획팀장, 전산감사의 승인하에서만 이루어지는가?

(2) 모든 프로그램 수정 요구문서 들이 순서적으로 번호가 매겨지고 보고되는가?

(3) 프로그램관련 문서(시스템, 운영, 사용자 문서)가 프로그램 수정을 반영하여 수정되는가?

(4) 컴퓨터 운영자가 프로그램 목록, 응용 시스템 문서에 접근할 수 없도록 되어 있는가?

(5) 응용시스템 프로그래머 그룹이 테스트한 프로그램을 실제가동환경으로 이동시켜서 설치하는 것이 금지되어 있는가?

(6) 응용시스템 프로그래머들의 시스템 프로그램 저장실(라이브러리) 접근이 금지되어 있는가?

(7) 프로그램 수정 후 프로그램의 코드내부에 특정계좌에의 불법 입금 등 범죄 유발 내용과 같은 불법적인 수행내용이 포함되어 있는지를 검사하는 방법이나 절차가 있는가?

(8) 시스템 변경이 있을 경우 운영 절차서는 신속히 변경되며 관리자의 승인을 얻고 있는가?

2) 데이터베이스 관리통제

(1) 사용자는 데이터, 레코드, 거래 등 업무에 필요한 부분만으로 접근이 제한되고 있는가?

(2) 분산처리환경에서 데이터 정의, 데이터 사전 등을 일치시키는 통제가 있습니까?

(3) 분산처리환경의 각 지역센타에서 하드웨어 운영과 데이터 보안에 대한 절차를 지키는지를 확인합니까?

(4) 데이터베이스에 저장되어 있는 응용시스템별 데이터에 대하여 다음과 같은 데이터 무결성 검증방법이 수행됩니까?(매칭검사, 밸런싱 체크, crossfooting, 순서검사, 중복검사)

(5) 응용시스템에 의해 처리되는 데이터의 완전성을 검증하기 위해 레코드 일련번호, 레코드의 수 및 통제합계를 사용합니까?

(6) 테이블 갱신의 경우 레코드의 변경 전후의 상태를 포함하여 저장됩니까?

(7) 원장을 갱신할 때 사용단말기 번호, 사용자 번호, 입력시간 등이 기록되어 잘못을 추적하도록 되어 있습니까?

(8) 원장변경 시에는 원장변경의뢰서가 작성되고 책임자 및 검사역의 서명이 반드시 있습니까?

2. 통제의 효과성

(1) 전산시스템의 전반적인 보안통제 수준에 대한 만족도는 어느 정도입니까?

(2) 전반적으로 보안통제의 유용성(조직목표달성, 전산시스템 목표달성,. 비용-효과성, 데이터 보호, 자원의 효율적 사용 등)은 어느 정도입니까?