델파이 XE2 DataSnap 서버의 보안성 개선을 위한 시큐어 코딩에 관한 연구

정명규[†], 박만곤^{††}

A Study on the Secure Coding for Security Improvement of Delphi XE2 DataSnap Server

Myoung-Gyu Jung[†], Man-Gon Park^{††}

ABSTRACT

It is used to lead to serious structural vulnerability of the system security of security-critical system when we have quickly developed software system according to urgent release schedule without appropriate security planning, management, and assurance processes. The Data Set and Provider of DataSnap, which is a middleware of Delphi XE2 of the Embarcadero Technologies Co., certainly help to develop an easy and fast-paced procedure, but it is difficult to apply security program and vulnerable to control software system security when the connection structure Database-DataSnap server-SQL Connection-SQL Data set-Provider is applied. This is due to that all kinds of information of Provider are exposed on the moment when DataSnap Server Port is sure to malicious attackers. This exposure becomes a window capable of running SQL Command. Thus, it should not be used Data Set and Provider in the DataSnap Server in consideration of all aspects of security management. In this paper, we study on the verification of the security vulnerabilities for Client and Server DataSnap in Delphi XE2, and we propose a secure coding method to improve security vulnerability in the DataSnap server system.

Key words: System security, Secure coding, Delphi, DataSnap

1. 서 론

오늘날과 같은 정보화 사회에서 아무리 강조해도 지나치지 않는 것이 “보안(Security)”이다. 작게는 개인의 정보에서부터 크게는 기업 또는 국가 전체의 기간 망을 뒤흔들 수 있는 것이 정보시스템의 악의적인 공격으로 인해 중대한 위기를 초래할 수 있다. 그만큼 사회의 발전에 대한 일조를 한 정보시스템이

이제는 위협의 대상으로 발전을 한 것이다. 2011년에는 해킹과 사이버 공격이 55억 건이며 이는 2010년에 비해 81%가 증가한 것이다. 최근에는 침입차단시스템과 보안장비를 우회하고 보안취약점을 악용하는 제로데이 공격, 해킹 등 사이버 공격으로 확산되는 추세이다[1,2]. 또한 특정 소프트웨어를 대상으로 보안취약점을 지능화된 기법을 이용해 공격하는 APT (Advance Persistent Threat: 기존 해킹과 달리 불특

※ Corresponding Author: Man-Gon Park, Address: (608-737) Yongso-Ro 45, Nam-Gu, Busan, Rep. of Korea, TEL: +82-51-629-6240, FAX: +82-51-628-6155, E-mail: mpark@pknu.ac.kr

Receipt date: Mar. 23, 2014, Revision date: May. 13, 2014
Approval date: May. 23, 2014

[†] Associate Member, Dept. of Information Systems, Pukyong Nat. Univ., Rep. of Korea
(E-mail: mgjung@gmail.com)

^{††} Life Member, Dept. of IT Convergence and Application Engineering, PuKyong Nat. Univ., Rep. of Korea

※ This work was supported by the Pukyong National University Research Abroad Fund in 2012 (PS-2012-020) and National Research Foundation (NRF) of Korea (2013-K1A3A1A09076037).

정 다수가 아닌 특정한 대상을 지정해 공격하며 오랜 시간 동안 잠복하면서 정보를 빼내는 게 특징) 공격이 확산되는 추세이다. 사이버 공격에서 약 75% (Gartner: 2005.05)가 소프트웨어 자체의 보안취약점을 찾아내어 이를 악용하는 것으로 웹사이트 공격이 대표적이다[3].

응용 소프트웨어 시스템의 보안 취약점으로 인해 공격을 받는 피해가 세계적으로 연 1800억불에 이르며, 소프트웨어가 인도와 설치 이후에 빠진 보안 취약점을 보완하는 것이 더 많은 비용이 지불되며, IBM의 통계에서는 초기 개발 시 보안요소를 확인하고 적용하는 것에 비해 30배에서 최고 100배 이상의 비용이 발생하는 점을 지적하고 있다[4-8].

대한민국 정부는 소프트웨어 개발 시 보안 적용을 체계적으로 수행할 필요가 있음을 인지하고 제도적으로 강구하기 시작했다. 행정안전부는 “소프트웨어 개발보안 가이드”와 “소프트웨어 보안약점 진단 가이드”를 제작하여 배포 하고, 사이버 공격의 가능성에 대한 잠재적 보안 취약점을 시스템 개발 초기단계부터 사전에 기획하고 제거하는 기법을 의무화 하도록 하였다. 2012년 12월부터는 국가 행정기관이 추진하는 시스템 개발비가 40억 이상의 정보화 시스템 사업에 먼저 적용하고, 그 대상을 점차 확대할 예정이다.

소프트웨어 개발을 위한 시큐어 코딩(Secure Coding)이란 사이버 공격에 대응하기 위하여 소프트웨어의 개발단계부터 보안 취약점을 분석하고 대비하여 공격 시도를 원천 차단한다는 개념이다. 구체적으로 말하면, 소프트웨어 개발과정에서 개발자의 실수나 논리적 오류 등으로 인한 소프트웨어의 보안 취약점을 최소화하고, 해킹 등의 보안위협에 능동적으로 대응할 수 있는 소프트웨어를 개발하기 위한 일련의 과정을 일컫는다. 소프트웨어 시큐어 코딩은 소프트웨어 개발 생명주기의 각 단계별로 요구되는 보안성 확보를 위한 보안 프로그램의 계획 및 관리 활동이라 할 수 있다[9-13].

보안 프로그램은 날로 늘어나고 수준이 높아가고 있는 해킹 및 악성코드를 개별적으로 분석해 대응할 수 있는 패턴을 개발하고 적용하는 방식을 개발하는 것이었다. 소프트웨어 시큐어 코딩은 개발단계에서 이러한 보안 취약점을 미리 보완하고 대처하기 때문에 사후 대응 방식의 보안 취약점 제거 비용 보다

수십 배의 비용을 절감할 수 있다고 보고되고 있다. 소프트웨어 개발단계에서 보안을 적용하고자 하는 시큐어 코딩(secure coding)의 연구는 현재 많이 알려져 있는 C++, JAVA 또는 웹 개발 언어의 프로그램 개발에 대한 시큐어 코딩에 관련한 것이 대다수이다[14-21].

본 논문에서는 시큐어 코딩 기법을 통해서 정보시스템 구축 시 사용되는 Embarcadero Technologies사의 Delphi XE2의 미들웨어인 데이터스냅에 대한 보안 취약점을 확인하고, 보안 취약성 개선을 위해서 데이터스냅 서버와 데이터스냅 클라이언트를 통하여 데이터 셋과 제공자는 사용을 하지 않고 시큐어 코딩할 수 있는 방법을 제안하고자 한다.

2. 관련연구

2.1 소프트웨어 보안 취약점

정보시스템의 보안을 제도적 접근으로 체계적으로 실행한 국가는 미국을 들 수 있으며, FISMA (Federal Information Security Management Act: 2002: 정보보안에 대한 미 연방정부의 기본법적인 역할), ITMRA (Information Technology Management Reform Act: IT관리혁신법: 1996), e-Government Act (전자정부법: 2002)의 내용이 보조적 장치이다[9]. 개발되는 소프트웨어의 보안 취약점은 Table 1과 같이 입력 데이터 검증 및 보안기능 등 일곱(7)가지의 유형으로 분류 할 수 있다.

최근에는 모바일 장치가 급속도로 발전하면서 이에 대한 보안 취약점을 분석하는 여러 가지 방법이 발표되고 있다. 구글사의 안드로이드 플랫폼에 Coverity사의 분석도구로 프로그램 소스코드를 분석한 결과 359개의 버그가 검출되었다. 이중에 88개의 버그는 보안 취약점 및 시스템 충돌을 유발할 수 있는 수준이었다[18]. 소프트웨어 소스코드에 대한 취약점 분석 도구는 개발자가 소스코드를 작성한 후 보안 취약점을 검사하기 위한 분석도구이다. Table 2는 개발자는 자신이 작성한 소스 코드에서 보안 약점을 진단하는 간편 방법이다.

2.2 시큐어 코딩(Secure Coding)

소프트웨어 개발과정에서 보안 약점을 완벽히 제거한다는 것은 사실상 거의 불가능하며 개발자의 실

Table 1. Type of Security Vulnerability

Type	Description
Vulnerable to input data validation	Missing or inadequate validation of the input values of the program, Security weaknesses caused by wrong data formatting (Examples: SQL Insert, Resources Insert, Cross-site Script, and so on)
Inappropriate security functions	Security vulnerability occurred when security functions are inappropriately implemented (Examples: inappropriate authorization permit, important information stored in the form of plain texts, hardcoded passwords, and so on)
Inappropriate management of time and status	The parallel system to support the simultaneous or almost simultaneous performing. Security weakness that should take place in management improperly and state time, In an environment where more than one process is running. (Examples: TOC. TOU, Recursive functions that do not use a control statement, and so on)
Insufficient error handling	Security weaknesses that arise when important information (systems, etc.) are included in the error information can be treated poorly or not processed the error. (Examples : Information disclosure through corresponding member, the error message on the status of error, and so on)
Code errors	Type conversion error, Security weaknesses that arise coding errors by developers such as inappropriate resource(memory, etc.) returns. (Examples : Null pointer Dereference, Release of inadequate resources, and so on)
Security vulnerability information when capsulized	Security weaknesses that possible to data leakage to unauthorized users when encapsulating critical data or functionality insufficiently. (Examples : Debugger code that remains without being removed, System data information disclosure, and so on)
API misuse	Security weaknesses that arise when use the API contrarily to the intended method of use or vulnerable API. (Examples : Security decision depends on Lookup, and so on)

Table 2. Diagnosis Methods of Security Vulnerability

Section	Content	characteristic
Static Analysis	Without running the software, Analysis of security vulnerabilities at the source code level (Mainly used in the software development process)	Software modification cost savings because of finding a security weakness in the early stages of software
		The integrated security weaknesses between components may be found in the restrictive
		Design, Structural weaknesses in the security point of view can not be found.
Dynamic Analysis	Analysis of software security weaknesses in the execution environment (Mainly used in the software testing phase)	There is no need to source code
		Improved coverage and accuracy
		It is affected by the level of the user of the tool
		It is not possible to discover vulnerable aspect of the structure

수, 논리오류 등 보안 약점을 최소화하고 보안 위협으로부터 안전한 소프트웨어를 개발하는 활동이 수반되어야 한다. 그래서 소프트웨어 개발 수명주기에

서 개발 단계에서 보안약점을 제거하는 코딩 활동을 시큐어 코딩(Secure Coding)이다. Table 3은 응용분야별 임베디드 소프트웨어 시스템들의 시큐어 코딩

Table 3. Coding rules of embedded software

Coding rules	Field
MISRA-C:2004	Manufacturing sectors (automotive, aerospace, communications, medical, defense) The purpose of improving the reliability of automotive software mainly
JSF(Joint Strike Fighter) Air vehicle C++ coding standard	Fighters, bombers, ground attack aircraft The purpose of improving the reliability of U.S. and British aircraft software mainly
CERT C	Applicable to all software written in C language

규칙들이다. 소프트웨어 개발 수명주기에서 시스템 분석, 설계, 개발(코딩), 테스트 및 보수유지로 이어지는 각 개발 단계별로 코딩 규칙에 대한 검사를 실시하며, 테스트 단계에서는 시큐어 코딩이 적절하게 준수되었는지에 대한 검토를 수행함으로써 소프트웨어 개발 완료 후에 투입되는 비용절감과 보안 취약 부분을 관리할 수 있기 때문에 소프트웨어 보안 취약성 감소효과를 기대할 수 있다[4].

M-SCSA(Mobile-Secure Coding Self Assessment)를 이용하여 모바일 응용 프로그램에 대한 시큐어 코딩의 보안수준을 스스로 평가하기 위한 도구로 사용이 되고 있다. 여기에 시큐어 코딩의 한 분야인 방어적 프로그램 기법 (타당하지 않은 입력으로부터 프로그램 보호, 프로그램이 실행될 때 스스로를 검사하는 코드, 오류 처리 기법, 예외, 공격적인 프로그래밍, 디버깅 보조도구 제거 등)을 적용할 수 있다 [10-13].

3. 델파이 XE2 DataSnap의 보안 취약점 확인

대규모 시스템 구축에서는 거의 3-Tier로 시스템이 구축되고 있으며, 소규모 프로젝트에서도 세 단계 (3-Tier) 시스템으로 구축되는 사례가 많아지고 있다. 이는 데이터베이스와 클라이언트 사이에 미들웨어를 두어 이 미들웨어에 필요한 비즈니스 로직을 넣어서 보안성을 더욱 강화하고 클라이언트는 가볍게 하여 정보시스템 운영을 원활하게하기 위함이다. 본 논문에서도 Fig. 1과 같이 세 단계로 테스트 시스템을 구축하였으며 데이터베이스는 마이크로소프트사의 MSSQL 서버를 사용하고 미들웨어는 델파이 XE2의 데이터스냅 서버를 사용하며 클라이언트는 델파이 XE2를 사용한다[13-15].

여기서 한 가지 먼저 정의를 하고 넘어가야 할 것이 있다. 그것은 다음과 같은 가정이 먼저 성립이 되

어야 본 논문에서 말하고자 하는 내용이 성립이 된다.

Fig. 1. Three-step test system.

[가정1] 악의적인 사용자가 특정기업의 공인 IP를 알아내었다.

[가정2] 또한 데이터스냅 서버가 사용하는 통신 포트 번호를 확보하였다.

위에서 언급한 가정은 악의적인 사용자가 특정 기업의 내부정보를 탈취하려거나 또는 정보의 탈취보다는 해당 기업의 정보 시스템을 무력화하기 위해 사전 준비 작업의 완료를 나타낸다.

3.1 데이터스냅 서버

클라이언트가 서버에 접속을 하여 데이터를 처리하는데 있어서 가장 먼저 델파이 XE2에서 데이터스냅 서버를 작성 하여 실행이 되어야 하는데 데이터스냅 서버의 작성 방법은 File -> New -> Other를 선택하면 새로운 프로젝트를 선택할 수 있는 창이 화면에 표시된다. 여기서 “Delphi Projects” 아래에 보면 “DataSnap Server”라는 항목이 보인다. 이 항목을 선택한 후 OK버튼을 클릭하면 데이터스냅 서버의 기본적인 파일을 생성해주는 마법사의 창이 나타난다. 프로젝트 파일을 생성하는 마법사의 단계는 아래 Table 4에서처럼 네(4) 단계로 이루어져 있다.

Table 4의 네 번째 단계에서 완료 버튼을 클릭하면 데이터스냅 서버의 기본 베이스 프로그램 단위들

Table 4. Four Steps of the Wizard of Data Snap servers

Step	Item	Contents
1	Select the application	The selections of an application function are divided into 1) VCL Foem, 2) Console, 3) Service.
2	Select the server function	Functions of the snap of server data in XE2 select 1) Protocols selection, 2) Authentication, 3) Server Methods class 4) Filters, 5) JAVA Scripts, 6) Mobile Connectors.
3	Select the server port	The XE2 a window for selecting a communication port used data snap server is based on 211 port. This paper is using this port.
4	Select the component	Finally, the data snap server will select the component.

이 표시된다. 세 번째 단계에서 선택하여 입력하는 통신 포트는 매우 중요하며 데이터스냅 서버의 기본 포트인 211 대신에 다른 포트 번호를 입력하여 사용할 것을 권고 한다. 또한 시스템에서 사용할 포트가 현재 다른 애플리케이션에서 사용하고 있는지를 “Test port” 버튼을 클릭하여 확인하고 없는 포트를 사용하여야 하며 정상적으로 사용할 수 있는 포트를 입력하였으면 “Test Port Succeeded”라는 메시지 창이 표시된다.

새로 생성된 서버의 단위 중 Fig. 2는 클라이언트와 서버간의 통신을 전담하는 유닛(Unit)이다. 서버에서 접속된 클라이언트의 상태를 확인하고 접속 세션(Session)을 관리한다. DSTCPServerTransport1은 서버에서 사용할 쓰레드(Thread)의 개수와 풀(Pool)을 사용할 것인지 버퍼(Buffer)의 크기(Size)는 어떻게 할 것인지를 설정한다.

DSServerClass1은 서버에서 작성되어 각 클라이언트에서 사용할 비즈니스 로직을 개발한 유닛(unit)을 연결하는 클래스로 각 메소드(Method) 하나에 하나의 클래스가 연결 되어야 한다. 또한 각 메소드 유닛(Method Unit)의 수명주기를 관리하는 기능도 있

Fig. 2. Container Unit of data snap server.

다. “Session”은 접속한 클라이언트가 접속할 시점에 생성이 되며 접속을 종료하면 해제가 이루어진다. 이렇게 함으로서 서버의 기본 토대가 마련되었으므로 실제 데이터베이스 서버를 설정하며 여기에서는 MS SQL 서버를 데이터베이스 서버로 사용을 한다. 본 논문에서는 보안에 대한 테스트를 진행하기 위해 메소드 유닛에 데이터베이스 컴포넌트를 둘 것이며, 데이터베이스 컴포넌트는 dbExpress의 TSQLConnection (“SQLConnection1”) 컴포넌트를 사용한다. 데이터베이스 컴포넌트를 설정하는 프로퍼티(Property)의 내용은 Fig. 3과 같다. 데이터베이스를 접속하는 드라이버(Driver)는 MS SQL이며 데이터베이스 명, 호스트 명, 사용자 명, 비밀번호 등을 설정한다.

많은 개발자들이 개발 시의 편리함과 관리를 원활

Fig. 3. SQLConnection Property.

히 하기 위해 데이터베이스 사용자 아이디(ID)를 “sa”로 설정을 하는 우를 범하고 있는데 이는 매우 위험하므로 반드시 해당 데이터베이스의 사용자를 생성하여 특정 데이터베이스만 사용을 할 수 있게 설정되고 프로그램 되어야 한다. 또한, 그림 3의 하단 부에 “TableScope” 프로퍼티가 있는데 이의 “tsSynonym”과 “tsSysTable” 항목은 기본 값은 “False”가 설정되어 있다. 본 연구에서는 이 두 개의 프로퍼티를 참(true)으로 설정을 하여 진행할 것이다. 이 두 가지 항목은 시스템 테이블에 접근을 허용하는 프로퍼티로 시스템 개발을 진행하다 보면 어쩔 수 없이 시스템 테이블을 접근하여 데이터를 처리하여야 할 경우가 발생하기 때문이다.

그리고 SQLConnection에 연결하여 데이터를 처리할 쿼리(Query) 컴포넌트와 제공자 컴포넌트는 각각 4개씩 갖다 놓고 사용을 할 것이다. 이름은 알기 쉽게 Table 5와 같이 설정한다.

Table 5. Connection of the provider component and query

Query component	Provider component
sqCOM	pvsqCOM
sqINS	pvsqINS
sqUPT	pvsqUPT
sqDEL	pvsqDEL

3.2 데이터스냅 클라이언트

클라이언트는 일반적인 VCL 애플리케이션을 선택하여 프로젝트를 생성하면 된다. 여기에 데이터베이스 접속 컴포넌트와 데이터스냅 제공자 컴포넌트를 가져다 놓고 둘을 연결한 다음 실행되어 있는 데이터스냅 서버의 접속 Port를 데이터베이스 컴포넌트에 입력을 한 후 연결을 참(true)으로 하면 접속이 완료된다. 클라이언트에서 데이터스냅 서버의 데이터베이스에 접속을 하기위한 데이터베이스 컴포넌트의 설정은 Fig. 4와 같으며, 드라이버의 선택을 “DataSnap”을 선택하면 접속할 수 있는 각종 프로퍼티 설정을 할 수 있게 나타난다. 여기서 앞서 작성한 데이터스냅 서버의 통신 포트를 “211”로 맞추어 주면 된다.

데이터스냅 클라이언트에서의 한 가지 중요한 작업이 있는데 이는 현재 구동중인 데이터스냅 서버의

Fig. 4. Datasnap client.

각종 메소드를 가져오는 것이다. 이것을 하지 않으면 서버에 접속은 가능하나 사용하여 할 메소드는 처리를 할 수 없다. 메소드 클래스(Method Class)를 생성하는 방법은 SQLConnection 컴포넌트를 오른쪽 마우스를 클릭하면 조그마한 팝업창이 뜨는데 이 팝업창의 내용 중 “Generate DataSnap client classes”를 선택하면 현재 실행되어 있는 서버의 메소드를 가져와서 사용할 수 있다. 이렇게 가져온 서버 메소드 유닛(Server Method Unit)을 프록시(Proxy) 유닛으로 저장을 하여 사용한다.

3.3 데이터스냅에서 보안 취약점 테스트 결과

SQLConnect와 제공자간의 연결과 제공자와 클라이언트 데이터 셋에 대한 연결을 한 후 데이터스냅 보안에 대한 테스트가 진행 된다. Fig. 5는 앞서 각 컴포넌트간의 연결을 한 직후 클라이언트 데이터 셋 컴포넌트의 제공자 프로퍼티를 선택한 설계 시점의 화면이다. 이처럼 각 컴포넌트간의 연결을 했을 뿐인데 데이터스냅 서버에서 사용하고 있는 데이터 셋의 모든 제공자가 표시 된다. 이것은 허가받지 않은 사람이 서버에 접속하여 해당 서버의 데이터 접근에 관련한 제공자를 모두 사용할 수 있는 상황이 되었다.

Fig. 6은 프로그램을 실행한 후 “select name from

Fig. 5. Provider List.

Fig. 6. Access of the user table of the data snap.

sysobjects where type='U'” SQL 명령을 실행한 화면으로 해당 제공자를 통하여 MS SQL 쿼리를 실행하면 그대로 사용이 가능한 예를 보여 준다. 만약, 악의적인 사용자가 “drop Table” 명령을 실행하면 아마도 해당 기업에서는 각종 업무를 처리하는 데이터베이스 서버의 테이블이 삭제되어 기업 전체의 업무가 마비될 수도 있다.

더 큰 문제는 사용자 테이블뿐만 아니라 시스템 테이블 까지 접근이 가능하다는데 있다. Fig. 7은 SQL의 시스템 테이블을 가져오는 명령을 실행한 화면으로 다른 제약조건 없이 허가 받지 않은 사용자가 시스템 테이블을 열람하고 있다. 입력된 SQL 문장은 “sp_tables” 으로 SQL 서버의 모든 테이블 즉, 시스템 테이블과 사용자가 로그인한 데이터베이스의 사용자 테이블을 모두 조회하는 스토어드 프로시저 (Stored Procedure) 명령이다. 시스템 테이블이 조회되는 것은 데이터스냅 서버에서 데이터베이스에 로그인 처리 시에 로그인 사용자가 “sa” 이기 때문이다.

다음은 데이터스냅 서버의 메소드에 관한 내용으로 데이터스냅 클라이언트 프로그램 작성 시 자동 (“Generate DataSnap client classes”)으로 생성한 프록시에 대한 분석이다. 메소드 유닛(Method Unit)인 프록시에 담기는 내용은 해당 데이터스냅 서버가

Fig. 7. Access of the system table of the data snap.

```
function TServerMethodsIClient.ReverseString(Value:
string): string;
begin
  if FReverseStringCommand = nil then
    begin
      FReverseStringCommand :=
        FDBXConnection.CreateCommand;
      FReverseStringCommand.CommandType :=
        TDBXCommandTypes.DSServerMethod;
      FReverseStringCommand.Text :=
        'TServerMethodsI.ReverseString';
      FReverseStringCommand.Prepare;
    end;

    FReverseStringCommand.Parameters[0].Value.SetWideString(
      Value);
    FReverseStringCommand.ExecuteUpdate;
    Result :=
      FReverseStringCommand.Parameters[1].Value.GetWideString;
  end;
end;
```

Fig. 8. An example of proxy ReverseString use.

사용하고 있는 모든 메소드에 대한 내용을 담고 있다. 그러므로 Proxy는 데이터스냅 서버에 적용된 상세한 비즈니스 로직의 구현 내용은 몰라도 그 구현에 접근할 수 있는 메소드 프로시저 (Method Procedure)는 확인이 가능하므로 이 프로시저에 해당 인수에 적절히 값을 넣어서 시스템을 괴롭힐 수 있다. Fig. 8은 Proxy 의 내용 중 입력된 문자를 뒤집어서 되돌려주는 ReverseString 에 대한 서버 메소드 (Server Method)이다. 예시에서 보는 바와 같이 서버에서 구현된 내용은 볼 수 없어도 해당 프로시저가 어떤 인수를 사용하는지 알 수가 있으며, 여기서는 “Value”라는 문자 인수를 입력받아서 처리된 결과를 문자 형식으로 되돌려 주고 있다.

Fig. 9는 위에서 실시한 데이터스냅에 대한 테스트를 실시한 결과를 그림으로 표시한 것이며 데이터스냅에서 제공자를 사용하여 시스템을 개발할 경우

Fig. 9. problem of the test result.

발생될 수 있는 보안 취약점을 나타낸 것이다. 여기서 보안 취약점은 데이터스냅 서버에서 제공하고 있는 모든 제공자 정보가 클라이언트에서 확인이 가능하여 제공자의 CommandText를 통한 악의적인 쿼리가 실행 가능하다.

또한, 자동으로 생성한 프록시에 대한 분석에서는 데이터스냅 서버가 사용하고 있는 모든 메소드 프로시저의 정보가 확인이 가능하여 공격자는 특정 프로시저에 해당 인수에 적절히 값을 넣어서 시스템을 괴롭힐 수 있다.

4. 데이터스냅 서버에서 시큐어 코딩 적용 방법의 제안

4.1 데이터베이스 사용자 (Database user)

많은 프로젝트에서 개발자의 편의를 위해 데이터베이스 시스템 전용 사용자 아이디인 “sa”를 프로그램에서 많이 사용을 하는데 이는 매우 위험함으로 DBA는 반드시 특정 데이터베이스만을 사용하게 하는 사용자 아이디를 별도로 부여하여 프로젝트를 진행하여야 한다. 데이터스냅 서버의 데이터베이스 컴포넌트의 프로퍼티 설정에서 “TableScope”가 있는데 기본 값 설정은 “tsSynonym”과 “tsSysTable” 항목이 “False”로 설정되어 있음을 확인하고 이를 변경하지 말아야 한다.

4.2 데이터 셋과 제공자(Data Set & Provider)

시스템 구축 시 개발일정에 맞추어 빨리 개발을 진행하고자 보안에 대한 충분한 검토가 이루어지지 않고 개발이 되면 자주 보안에 취약한 부분이 발생된다. 데이터스냅에서의 데이터 셋과 제공자는 분명 개발을 쉽고 빠르게 진행하는데 도움이 주지만 Database-DataSnap server - SQL Connection - SQL Data set - Provider로 연결이 되는 구조에서는 보안 프로그램의 적용이 어렵다. 이는 데이터스냅의 통신 포트가 확인되는 순간 모든 제공자의 정보가 노출되기 때문이며, 이 노출은 SQL 명령을 실행할 수 있는 창구가 되는 것이다. 그러므로 데이터스냅 서버에서 데이터셋의 제공자는 사용을 하지 않는 것이 보안성 측면에서는 더 안전하다.

4.3 메소드 (Method)

```
function TServerMethods1.doCheckUser(aUserID, aUserPW: String): Boolean;
begin
    Result := False;
    if (not isCheckLogin) or (isSession = 0) then Exit;
    try
        USQL := 'SELECT LID, LNM, LDEPT FROM User WHERE LID = ''' + Trim(aUserID) + '''';
        sqExec.Close;
        sqExec.CommandText := Trim(USQL);
        sqExec.Open;
        if not sqExec.Eof then
            begin
                Result := True;
            end;
    except
        on E: Exception do
            begin
                raise Exception.Create(E.Message);
            end;
    end;
end;
```

Fig. 10. An example of method procedure use.

앞의 내용에서 데이터셋과 제공자를 사용하지 않으면 어떻게 시스템 구축을 하여야 하는지 의문이 있을 수 있다. 이를 해결하는 방법이 메소드 프로시저(Method Procedure) 만 사용하여 서버 시스템을 개발 하여야 한다. 메소드만으로 프로그램을 하더라도 각각의 메소드는 다양한 인수를 입력 받을 수 있고, 다양한 결과 값을 돌려 줄 수 있으므로 가능하다. Fig. 10의 예에서처럼 참 또는 거짓을 돌려주는 메소드이지만 여기에 문자열(String), 숫자(Integer) 및 쿼리 실행 결과(Data Set)와 JSON (JavaScript Object Notation) 타입 등 다양한 결과 값을 돌려 줄 수 있다.

데이터스냅 서버의 메소드를 구현할 때 Fig. 10의 네 번째 라인에서처럼 로그인 확인과 세션 (정상적인 로그인이 이루어 졌을 경우 시스템에서 부여한 세션 ID)의 프로퍼티를 두어 모든 메소드에서는 이 두 가지를 항상 확인하여 정상적인 사용자에게만 프로시저를 실행할 수 있게 프로그램 되어서 보안성 취약점을 개선시킬 수 있다.

데이터스냅 서버에서 보안 취약점을 제거하고 구축하는 정보시스템의 안전성을 확보하기 위한 방법으로 Fig. 11과 같은 방법을 제안한다.

5. 결론 및 향후 연구

소프트웨어 시큐어 코딩(Secure Coding)이란 악성코드나 해커들의 공격에 대응하기 위하여 소프트웨어의 개발단계부터 보안 취약점을 분석하고 대비

Fig. 11. The proposed method.

하여 공격 시도를 원천 차단하는 방법이다. 이는 소프트웨어 개발과정에서 개발자의 실수나 논리적 오류 등으로 인한 소프트웨어의 보안 취약점들인 버그, 결함 또는 로직 결점들과 같은 소프트웨어 오류들을 최소화하고, 해킹 등의 보안위협에 능동적으로 대응하는 일련의 과정을 일컫는다.

소프트웨어 시큐어 코딩은 소프트웨어 개발 생명 주기의 각 단계별로 요구되는 보안성 확보를 위한 보안 프로그램의 계획 및 관리 활동이라 할 수 있다. 소프트웨어 산업에서 많은 업체들이 정보 시스템 분석, 설계 및 개발 단계에서 보안 취약점을 관리하지 않고 있으며, 일부에서 개발이 완료된 후 모의 해킹이나 보안 취약점에 대한 점검을 수행하고 있는 실정이다. 소프트웨어 개발단계에서 보안 관리 프로그램을 적용하고자 하는 시큐어 코딩(secure coding)의 연구는 현재 많이 알려져 있는 C++, JAVA 또는 웹 개발 언어의 프로그램 개발에 대한 시큐어 코딩에 관련한 것이 대다수이다.

본 논문에서는 데이터스냅의 클라이언트 및 서버에 대한 보안 취약점을 확인하고 이를 보완할 수 있는 시큐어 코딩 방법을 제안하였다. 이 제안 방법은 메소드 만을 사용하여 데이터스냅 서버 시스템을 세 단계로 구축 시에는 서버의 비즈니스 로직 개발 담당자의 부하가 가중되는 현상이 발생하며, 비즈니스 로직 개발자와 클라이언트 개발자간 의사소통이 적절히 이루어지지 않아 안정적인 시스템 구축의 방해요소가 될 수 있으며, 이로 인하여 메소드만을 사용하여 데이터스냅 서버 시스템을 3-티어로 구축 시에는 많은 개발 시간과 비용투자가 되지만 보다 안정적이고 보안이 잘 적용된 시스템 구축을 위해서는 반드시 필요한 작업이다.

향후에는 데이터스냅 서버의 REST에 대한 연구

도 필요하다. 이는 데이터베이스에 접속하는 프로그램 및 장치들이 다양하게 존재를 하고 있으며, 이들 또한 데이터스냅 서버에 접속하여 자료를 확인하고 데이터를 처리 할 것이다.

REFERENCE

- [1] Ministry of Security and Public Administration, *Software Development Security Guide*, 2012.
- [2] Ministry of Security and Public Administration, *Software Security Vulnerability Diagnosis Guide*, 2012.
- [3] Advance Persistent Threat Attack, http://navercast.naver.com/contents.nhn?rid=122&contents_id=32568(accessed Oct., 20, 2013).
- [4] J. Grembi, *Secure Software Development: A Security Programmer's Guide*, Cengage Learning Publishing Co., Singapore, 2008.
- [5] M. Dowd, J. McDonald, and J. Schuh, *The Art of Software Security Assessment: Identifying and Preventing Software Vulnerabilities*, Addison-Wesley Professional, Boston, 2006.
- [6] G. McGraw, *Software Security: Building Security In*, Addison-Wesley Software Security Series, Boston, 2005.
- [7] M.G. Graff and K.R. van Wyk, *Secure Coding: Principles and Practices*, O'Reilly Media, Sebastopol, 2003.
- [8] K.S. Hoo, A.W. Sudbury, and A.R. Jaquith, Tangible ROI through Secure Software Engineering, *Secure Business Quarterly*, Vol. 1, Issue 2, pp.1-6, 2001.
- [9] R. Seacord, "Secure Coding Standards," *Proceedings of the Static Analysis Summit*, pp. 14-16, 2006.
- [10] T.S. Lee, K.W. Lee, D.H. Won, and N.J. Park, "Information Technology Security Evaluation Using CERT C Secure Coding Standard," *Lecture Notes in Computer Science*, Vol. 7105, pp. 335-342, 2011.
- [11] M. Johns, C. Beyerlein, R. Giesecke, and J.

- Posegga, "Secure Code Generation for Web Applications," *Lecture Notes in Computer Science*, Vol. 5965, pp 96-113, 2010.
- [12] R.L. Jones and A. Rastogi, "Secure Coding: Building Security into the Software Development Life Cycle," *Journal of Information Systems Security*, Vol. 13, No. 5, pp. 29-39, 2004.
- [13] Delphi, <http://www.embarcadero.com/>(accessed Oct., 20, 2013).
- [14] Delphi XE DataSnap Development Courseware Manual, Embarcadero Technologies, <http://www.ebob42.com/courseware/>(accessed Feb., 12, 2014).
- [15] RAD Studio in Action - DataSnap 2010 white paper, http://www.embarcadero-info.com/in_action/radstudio/db.html (accessed Feb., 12, 2014).
- [16] D.H. Jung, J.Y. Choi, and S.H. Lee, "Nuclear-Related Software Analysis Based on Secure Coding," *Journal of The Korea Institute of Information Security & Cryptology*, Vol. 23, No. 2, pp. 243-250, 2013.
- [17] Y.H. Choi and E.M. Choi, "Analysing Weak Point of Android Applications by using Static Analysis based on Anti-patterns," *Journal of KIISE : Computing Practices and Letters*, Vol. 38, No. 2, pp.132-135, 2011.
- [18] D.W. Kim and K.H. Han, "A Study on Self Assessment of Mobile Secure Coding," *Journal of The Korea Institute of Information Security & Cryptology*, Vol. 22, No. 4, pp. 901-911, 2012.
- [19] S.M. Lee, J.S. Oh, and J.Y. Choi, "Comparative Analysis on Potential Error-Possibility and Security Vulnerability in Software," *Korea Computer Congress 2012*, Vol. 37, No. 1, pp. 106-109, 2010.
- [20] M.H. Kim and M.G. Park, "A Study on the Software Fault Modes and Effect Analysis for Software Safety Evaluation," *Journal of Korea Multimedia Society*, Vol. 15, No. 1, pp. 113-130, 2012.

- [21] S.K. Kim and J.I. Lee, "Analyzing Secure Coding Initiatives: An Ecosystem Approach," *Journal of The Korea Institute of Information Security & Cryptology*, Vol. 22, No. 5, pp. 1205-1216, 2012.

정 명 규

2012년 2월 부경대학교 (정보시스템전공) 공학석사
2014년 4월 부경대학교 대학원 IT융합응용공학과 정보시스템전공(박사과정 재학 중)

2005년~현재 인소프트 대표

관심분야: 소프트웨어 안전성 및 보안성, 공장자동화, 임베디드 소프트웨어

박 만 곤

경북대학교 수학교육(이학사)
경북대학교 전산통계학(이학박사)
Philippine Women's University (국제행정학석사)
University of Rizal System, Philippines(명예 기술학박사)

Dept. of Electrical and Computer Engineering, University of Kansas (Post Doc.)

1981년~현재 부경대학교 IT융합응용공학과 교수

2008년~현재 한국멀티미디어학회(KMMS) 회장 및 명예회장

2002년~2007년 정부간 국제기구 CPSC (콜롬보플랜기 술교육대학) 총재 (Director General and CEO)

2004년~2007년 Asia-Pacific Accreditation and Certification Commission (아태지역 인증 및 검증위원회) 위원장

2005년~2007년 유네스코 (UNESCO-UNEVOC) 자문위원, 아시아개발은행 자문관

관심분야: 소프트웨어 공학 및 재공학, 소프트웨어 신뢰성공학, 소프트웨어 안전성 공학, 비즈니스 프로세스 재공학(BPR), 멀티미디어정보처리기술, 정보시스템 성능평가 기법 및 도구, ICT-based HRD System