SAP GUI의 SW보안을 위한 ABAP 프로그래밍

임지은 최진영
고려대학교 컴퓨터정보통신 대학원 소프트웨어 공학과
dlawml79@korea.ac.kr, choi@formal.korea.ac.kr

ABAP Programing for SW security of SAP GUI

Jieun Im, Jinyoung Choi
Graduate School of Korea University Computer information communication

요 약

최근 급속도로 정보화 사회가 진행되어 가면서, 정보 보안에 대한 문제가 생겨났다. 많은 프로그램이 있지만 그 중 SAP사에서 많이 보편화 된 SAP GUI의 보안에 대하여 연구한다. SAP GUI에서 사용되는 언어 중 SAP에서만 사용하는 언어인 ABAP을 사용하는 경우 SQL Injection에 대한 문제점과 해결할 수 있는 방법을 논하고, 장단점을 논한다.

1. 서 론

정보화 시대가 되면서 정보의 보안은 점차적으로 중요하게 부각되고 있는 추세이다. ERPScan에서 발표된 자료에 의하면, 2012년 SAP(System, Application and Products)에서도 정보 보안에 대한 문제가 생긴 적이 있다. 그 이후로 SAP 내부에서는 보안에 대한 많은 관심을 가지게 되었다.

SAP GUI(Graphic User Interface)는 SAP사에서 가장 많이 보편화 되어있는 프로그램이다. 이 프로그램에서 지원해주는 언어는 C/C++, Java, HTML 그리고 ABAP/4가 있다.

이 중에서 SAP R/3에서만 사용하는 ABAP/4코딩 중 어떤 부분이 보안에 취약성을 보이는지 그리고 어떤 방식으로 코딩을 해야 보다 더 안전한 형식으로 보안을 할 수 있게 되는지에 대해서 논하고자 한다.

2. 정적 분석 도구 및 관련 연구

보안상의 문제에 관련해서 제공해주는 정적 분석 도구와 관련된 연구에 대해 논하여 보고자 한다.

2-1. CCMS(운영모드와 인스턴스 유지 보수)

[그림2-1] SAP ERP BASIC 어플리케이션 기반 구조 - IMG와 조직 관리 모델

CCMS(Computing Center Management System)은 SAP에서 제공하는 최소한의 내장된 바이러스 보호 기능을 제공한다.

이것은 시스템 프로파일과 시작 프로파일에 대한 접근을 관리한다. 이것은 경고 모니터 역할을 하면서 운영 시스템과 잠금 관리, 시스템 로드 활동, ABAP/4 프로그램 이상 종료에 대한 세부사항을 제공한다[1].

2-2. 아밥 소스코드의 코드 검사를 수행하는 코드검사 수행시스템 특허

아밥 소스코드의 코드 검사를 수행하는 코드검사 수행 시스템 특허는 최근 soft4soft라는 회사에서 제공하는 아밥(ABAP/4)의 코딩에 관련된 정적 분석 도구이다.

이는 시스템 고장 및 정지, 데이터 손실 및 변형, 과대한 성능 저하, 기능의 틀린 행동 및 결과, 복잡한 코드 및 철자 오류 등의 코딩 결함을 검사하여 아밥 어플리케이션의 유지보수성, 성능성, 정확성 및 보안성들을 신뢰할 수 있도록 한다[2].

이중 보안 취약점에 관련된 코딩 규칙 부분만 가지고 온다면 아래와 같다.

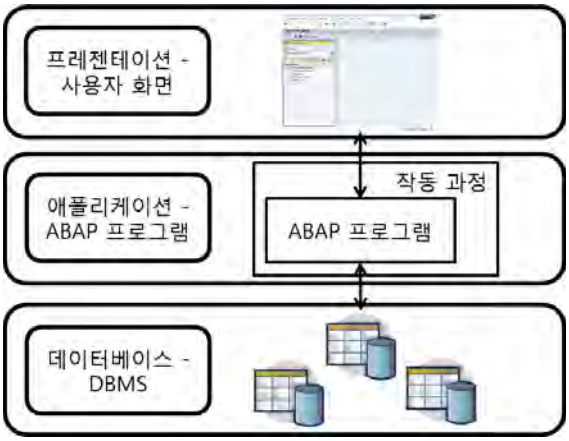
- Call of System Function
  (시스템 함수는 사용하지 말아야 한다.)
- Call of Transactions
  (Call Transaction 문장은 사용하지 말아야 한다.)
- Use of Database Hints
  (Select 문장 사용시 hints 구문을 사용하지 말아야 한다.)

3. ABAP/4 소개

SAP R/3(System, Application and Products Real-time 3-tier)는 SAP사에서 개발한 소프트웨어로서 현재 전 세계적으로 많이 사용되고 있는 ERP(Enterprise Resource Planning, 전사적 자원 관리 시스템)시스템

이다.

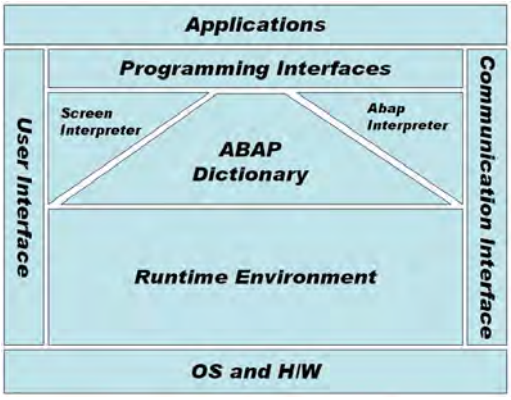
SAP R/3(System, Application and Products Real-time 3-tier)의 아키텍처(Architecture)는 [그림 3-1]과 같이 기본적으로 분산 환경의 3-tier모델이다.



[그림 3-1] SAP R/3 3-Tier 구조

SAP GUI를 통해 사용자에게 인터페이스를 제공하는 Presentation, 데이터를 저장하는 Database Server, 그리고 데이터 갱신이나 Dialogue등의 실제 서비스를 담당하는 Application Server등 3개의 계층구조로 구분되어 있다[3]

SAP R/3는 BASIS(ABAP + DATABASE), Core Module(FI, CO, MM, SD, PP등), 애드온(Add-On)으로 나뉘어져 있다. ABAP/4(Advanced Business Application Programming for 4-generation Language)는 SAP R/3의 애드온(Add-On) 프로그램을 개발하기 위한 4세대 언어이다[4].

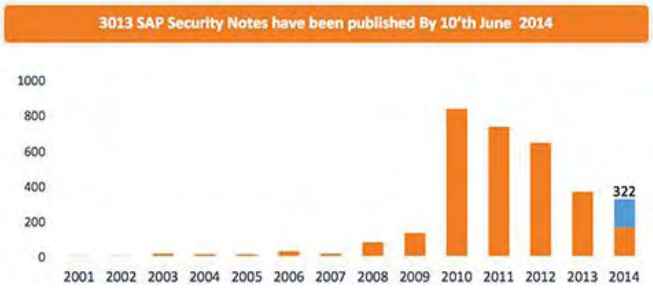


[그림 3-2] SAP R/3 시스템구조 안의 ABAP

이는 다양한 테스트, 모니터링, 디버깅 등의 베스트 툴로 통합된 워크벤치를 제공함으로써, 개발자가 단기간에 신뢰성 있는 정보시스템 제공이 가능하도록 지원한다[5].

이와 함께 개발자가 참조할 수 있는 SAP 참조 모델을 제공해 주고 있다. 이는 R/3 구조의 이해와 적용에 대한 판단을 하기 위해서 프로세스, 기능, 필요한 정보, 데이터 처리 등을 할 수 있는 각종 화면이 구비되어 있다[6].

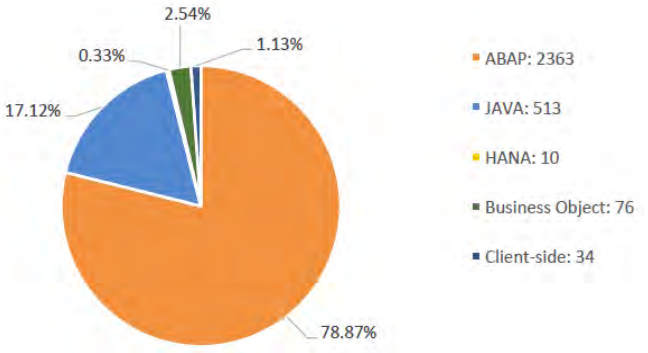
4. ABAP의 Security 상의 문제



[표 4-1] SAP Security Notes have been published. ERPScan 2014

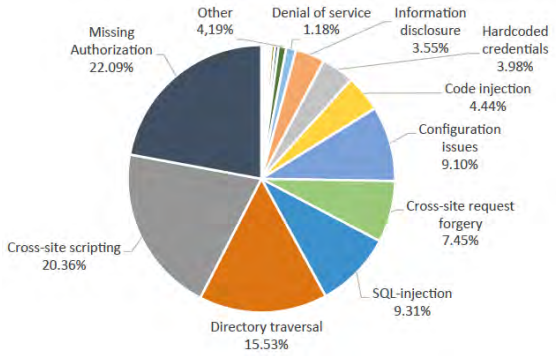
위 [표4-1]는 ERP Scan이라고 하는 SAP Security에 관련된 업체에서 발표한 내용이다.

표에서는 2012년 보안에 관련된 문제가 대두됨을 알 수 있다. 이 후 SAP에서는 보안에 관련된 부분에 관심을 가지고 해결하기 위해서 노력하고 있다.



[표 4-2] SAP Security Notres sorted by stacks. ERPScan 2014

위 표[4-2]는 2014년 발표된 자료로 어떤 부분에서 보안적인 문제가 일어났는지에 대한 통계이다. 이 자료를 토대로 한다면, 가장 많은 비중을 차지하고 있는 부분이 ABAP/4이다.



[표 4-3] SAP Security Notes sorted by type in SAP NetWeaver ABAP engine. ERPScan 2014

위의 [표 4-3]를 보면 보안상 어떤 부분에서 많이 경고를 받는지에 대해서 알 수 있다.

누락 승인(Missing Authorization)이란 말 그대로

허가를 받아야 하는 부분임에도 불구하고 지나쳐 갔다는 것이다. 이 부분은 보통 SAP 내부에서 개발된 프로그램으로 모니터링이 가능한 부분이다.

크로스 사이트 스크립팅(XSS)은 웹 애플리케이션에서 많이 나타나는 취약점의 하나로 웹사이트 관리자가 아닌 이가 웹 페이지에 악성 스크립트를 삽입하는 취약점이다. 주로 여러 사용자가 보게 되는 전자 게시판에 악성 스크립트가 담긴 글을 다시 게시판에 올리는 형태로 이루어진다[7].

디렉터리 접근 공격(Directory traversal)은 웹 서버와 연동되어 있는 경우에 발생하게 되는 위협인데, 웹 서버 설정상의 오류 혹은 중요 파일의 위치 오류들을 활용하여 직접 해당 디렉터리에 접근하여 자료를 변조 혹은 유출하는 방법이다.

SQL(Structured Query Language) Injection이란, 정상적인 사용자 입력을 우회하여 데이터베이스에 직·간접적으로 공격자가 SQL 질 의문을 삽입하여 원하는 데이터를 추출하는 공격방법으로 SQL 삽입이라고도 하며 보안약점 중 입력데이터 검증 및 표현의 한 기법으로 분류하기도 한다[8].

5. ABAP SQL Injection Attack

4에서 알아본 것과 같이 많은 보안 취약점이 있다. 이중 우리가 ABAP을 사용해서 보안의 취약점을 막을 수 있는 것은 SQL Injection 부분이다.

SAP Korea에서 제공하는 SAP help 자료에 따르면 SQL Injection은 두 가지 종류가 존재한다. SQL 조작(Manipulation) 공격에 대해서 알아보면 아래와 같다.

```
Original SQL statement:
    SELECT fieldlist
      FROM table1
    WHERE field = 'userinput'.
Examples for SQL injection attack:
    SELECT fieldlist
      FROM table1
    WHERE field = 'UNION ALL SELECT other_field
              FROM other_table WHERE '='.[9]
```

이 외에도 하나 더 있는데 그것은 아래와 같다.

```
Original SQL statement:
    SELECT fieldlist
      FROM table
    WHERE field = 'userinput'.
Examples for SQL injection attack:
    SELECT fieldlist
      FROM table
    WHERE field = 'anything' OR 'x'='x''.
    SELECT fieldlist
      FROM table
    WHERE field = 'x' AND email IS NUL; --'.[9]
```

이제 Code Injection 공격에 대해서 알아보면 아래와 같다.

```
Original SQL statement:
    SELECT *
      FROM table
    WHERE name = 'userinput'.
Examples for SQL injection attack:
    SELECT *
      FROM table
     WHERE name = ' a'; DROP TABLE users; SELECT * FROM
                    table1 WHERE name = '%''. [9]
```

6. ABAP SQL Injection 해결책

5에서 알아 보았듯이 보안 상의 문제가 있다. 이제 이런 문제에 대한 해결책에 대하여 논해보고자 한다.

6·1. RFC(Remote function call) 사용

RFC는 SAP에서 제공하는 외부 통신 프로토콜이다.

이것의 특징은 CPI-C 통신이라는 특수한 방식을 이용하여 외부에서 들어오는 정보에 대한 보안을 할 수 있다는 것이다.

RFC는 같은 System 내에서도 호출이 가능하지만, System 외부에서도 호출이 가능한 함수이다. 이 기능을 사용하면 처음에 Import 받는 Field 혹은 Table의 type의 정의 할 수 있다. 그로 인해 String으로 들어오는 SQL injection attack을 방어 할 수 있다.

하지만 차후 필드의 type이 변경이 될 경우 코드가 밀려서 깨질 수도 있다. 유지보수를 하게 될 경우 하나하나 변경을 하여야 하는 불편함이 있다. 만일 외부 타 프로그램과 연동 하였을 경우 외부 프로그램에도 영향을 주기 때문에 ABAP 개발자뿐만 아니라 외부 프로그램 개발자도 같이 변경해주어야 한다.

6·2. Field나 Table을 사용

Field symbol을 대신하여 Field나 Table을 사용하는 방법이다. Field symbol이란 대용량 업데이트 혹은 이송(Migration)시에 사용하는 방법이다.

Field symbol은 동적으로 필드를 설정해야 할 수 있다는 장점이 있다. 하지만 그로 인해서 데이터 보안에는 취약하다. 그렇기에 그것을 대신하여 Field를 설정하거나 Table로 받는 방법이 있다. 이 방법의 단점은 데이터를 너무 오래 잡아 Runtime Error를 유발시킬 수 있다는 점이다.

6-3. 웹 디버깅 툴을 사용 공격을 예측

피 들러(Fettler)같은 웹 디버깅 툴을 사용하여 클라이언트(컴퓨터)에서 서버로 요청한 내역과 결과를 확인 하여 공격을 예측하는 방법이 있다.

이것은 웹 브라우저를 연동(Interface)를 함으로써 인해 받을 수 있는 모든 공격에 유용하게 사용 될 수 있다는 장점이 있다.

하지만 이것은 공격이 들어오기 전에 미리 예측 할 수 없는 방법이라는 단점이 있다. 이것은 SAP GUI 쪽의 방법이라기 보다는 모든 연동되는 프로그램들의 문제를 해결해 줄 수 있을 것으로 생각된다.

6-4. 변환 함수를 사용

SAP GUI에서 제공하는 변환 함수를 사용하여 받은 코드를 한번 더 변환하는 방법이다.

그 예로는 CALL FUNCTION 'OBJECT_KEY_GET_KS'와 같은 함수가 있다. 이 함수는 오브젝트 넘버를 받아서 코스트 센터가 어디인지 알 수 있도록 해준다. 즉, 오브젝트 넘버를 넣으면 코스트 센터 넘버로 바꿔주는 것이다.

이러한 방법은 SAP 내부에서 제공하는 함수이기 때문에 따로 작업 할 것이 없다는 것이 장점이다. 하지만 SAP에서 제공해주는 것 이외의 것들은 따로 만들어야 하기 때문에 필요한 것이 없을 경우 따로 개발해야 하는 단점이 존재한다.

6-5. Class를 사용

함수와 비슷한 개념인 Class는 함수와는 다르게 System 내부에서만 사용할 수 있는 방법이다.

그 외에는 함수와 같이 Class를 만들어서 코딩을 해놓는 방식이다. 그렇게 되면 내부에서 같은 코딩을 여러 번 하지 않고 Class를 부르면 사용할 수 있게 된다. 그렇기 때문에 유지보수를 좀 더 효율적으로 할 수 있다. 하지만 Class를 사용하면 사용할 때 마다 필요한 부분뿐만 아니라 안에 개발되어 있는 전체를 불러와야 한다는 단점이 있다.

6-6. Lock object 사용

Lock object는 SAP에서 기본적으로 제공해주는 유저의 사용을 제한을 주는 프로그램이다.

이 방법은 함수와 같이 만들어서 프로그램에 적용하여 사용하는 방법이다. 이것의 장점이자 단점은 특정 유저만 들어 갈 수 있도록 설정할 수 있다는 것이다. 보안상으로는 좋지만 프로그램의 유지보수 시에는 특정 유저만 들어 갈 수 있도록 해 놓았기에 특정 유저가 아니면 수정 할 수 없다는 점이 있다.

7. 결 론

7-1 연구 결과

5에서 알아본 것과 같이 기존의 SQL Injection을 사용하였을 때 보안상의 문제만 제기 되었을 뿐 해결책은 제시되지 않았다.

그럼으로 6에서 여섯 가지 보안하는 방법을 제시하였고 서로의 장단점에 대해서 서술하였다. 이것을 효율적으로 사용한다면 보안상의 문제를 해결할 수 있을 것이다.

또한 RFC 혹은 Class를 사용한 보안 방법을 사용하게 될 경우, 한곳에서만 부르는 것이 아닌 여러 곳에서 동시에 부를 수 있기에 더욱 효율적으로 사용될 수 있다. 즉, 차후 유지보수나 튜닝을 하게 되면 여러 개의 프로그램이 아닌 하나의 함수 혹은 Class를 수정하면 된다는 장점이 있다.

7-2 향후 연구 방안

현재는 SQL Inject의 보안 상의 문제점과 이에 관련된 내용에 대해서 실질적인 코딩에 대한 부분이 들어와 있지 않다. 차후 이런 부분에 대해서 보안할 예정이다.

현재 다루지 않은 보안 상의 문제 점도 그렇다. 차후 지금은 다룬 SQL Injection에 관련된 것뿐만 아니라 보안에 관련 되어있는 다양한 문제에 대해서 연구해 보고 다른 언어와의 차이점에 대해서 연구할 예정이다.

8. 참고문헌

[1] ISACA, “Security, Audit and Control Features SAP ERP, 3rd Edition 한글판”, 사단법인 한국정보시스템감사통제협회, 2011.6

[2] 아밥 소스코드의 코드 검사를 수행하는 코드검사 수행 시스템, http://www.google.com/patents/WO2011122724A1?cl=ko

[3] 김성준, “Easy ABAP Programming”, 프리렉, 2008.

[4] 성승현, “SAP R/3 상에서 실시간 거래처 명세서를 위한 User-Exit의 효율적인 활용 방안”, 서강대 정보통신대학원, 2003.8

[5] Shahabi & Cyrus, “A Probe-Based Technique to Optimize Join Queries in Distributed Internet Databases”, Knowledge and information systems, 2000.

[6] CJ 시스템 SAP ERP 연구회, “실전 프로그래밍 ABAP/4 Programming Guide”, 성안당, 2008.4

[7] 이지선, “크로스 사이트 스크립팅 예방을 위한 정규식 기반 보안코딩 기법에 관한 연구”, 고려대 컴퓨터 정보통신대학원, 2012.12

[8] 한진규, “AST를 활용한 SQL Injection 취약점 예방 방법에 관현 연구”, 고려대 정보보호대학원, 2013.6

[9] SAP help, http://help.sap.com