원격근무를 위한 웹 기반 업무 시스템의 보안 요소 분석

남건우*, 김상천*, 박중길*

*국가보안기술연구소

Security Analysis on Developing Web-based Business System for Teleworking User

Nam Gun-woo*, Kim Sang-Chun*, Park Joong-Gil*

*National Security Research Institute

요 약

업무 편의성 요구 증가와 웹 서비스의 발전으로 웹 기반 원격근무 업무 시스템이 증가하고 있는 반면, 웹 취약점의 확산과 개방형 환경으로 인해 잠재적 위협이 증가되고 있다. 이러한 위협에 적절히 대응하기 위해서는 개발 단계 전반에서 보안 요소를 고려하여 안전한 업무 시스템을 구축해야 한다. 이를 위해 본 논문에서는 업무 시스템이 요구하는 보안 수준에 부합하는 보안 목표를 설정하고, 요구 보안 수준을 충족할 수 있도록 전체 개발 프로세스 상의 보안 관리 활동을 정의한다. 또한, 신원 인증 및 접근제어, 세션 관리, 이벤트 로깅, 데이터 검증, 알려진 취약점 대응, 그리고 프라이버시 보호와 같은 보안 요소들을 분석하여 시스템의 안전한 구축 방법을 제시한다.

I. 서론

인터넷과 웹 환경의 확산으로 인해 웹을 통한 정보시스템의 구축과 이를 통한 정보서비스의 제공이 보편화되고 있다. 원격근무를 위한 업무 시스템의 경우, 사용자 편의성의 향상과 유지 보수 비용 절감을 위해 점차 웹 기반으로 신규개발 또는 전이되고 있다. 그러나 웹은 인터넷/인트라넷 상에 공개되기 때문에 많은 위협에 노출되어 있다.[1] 원격근무를 위한 업무 시스템에 가해지는 위협에 적절히 대응하기 위해 현재 운영되고 있는 시스템의 취약점을 분석하여 대응할 수도 있지만, 이는 여러 면에서 어려움이 있다.

첫째, 운영중인 시스템의 취약점 분석 및 대응을 위해서는 대부분 응용프로그램의 수정 및 재개발이 필요하게 되고 이를 위해서는 많은 비용과 시간이 소요된다.

둘째, 현재 운영중인 시스템의 경우 취약점이 발견되어도 적절한 조치를 하는 것이 불가능할 때가 종종 존재한다.

셋째, 웹 응용프로그램의 수정이 빈번히 일어나는 경우 취약점 점검과 대응이 수시로 이루어져야 한다.

이러한 한계를 극복하기 위해서는 웹 응용프로그램 개발 단계에서 보안 요소를 분석하여 요구 보안 수준을 달성할 수 있도록 전체 개발 과정을 관리해야 한다.

이를 위해, 요구 수준에 부합하는 보안 목표를 설정하고 개발 프로세스 상의 보안 관리를 위해 필요한 활동을 정의한다. 그리고, 신원 인증 및 접근제어, 세션 관리, 이벤트 로깅, 데이터 검증, 알려진 취약점 대응, 그리고 프라이버시 보호 등 고려되어야 할 보안 요소에 대해 기술한다.

II. 본문

1. 보안 목표 파악

1) 위협 식별

웹 응용프로그램의 보안 요소를 고려하기 위해서는 웹 응용프로그램에 가해질 수 있는 위협을 정확히 파악하는 것이 중요하다.[2]

- 웹 응용프로그램과 연관된 자산과 해당 자산의 소유자 및 가치를 파악
- 이러한 자산에 대한 위협을 식별
- 식별된 위협이 자산에 미칠 수 있는 영향을 기밀성, 무결성 및 가용성의 관점에서 분석

2) 보안 목표 분석

웹 응용프로그램에 가해질 수 있는 위협 식별 후 개발될 웹 응용프로그램의 보안 요구 수준을 검토한다. 원격 근무를 위한 웹 기반 업무 시스템의 경우 다음과 그림 1과 같이 보안 목표를 설정할 수 있다.

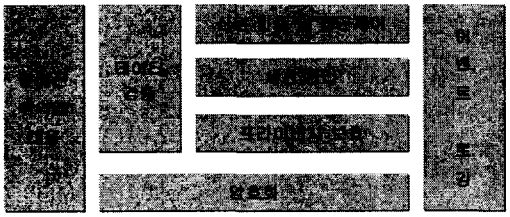

그림 1: 보안 목표 분석

2. 개발 프로세스 상의 보안 관리

웹 응용프로그램의 보안 수준을 보장하기 위해서는 그림 2와 같이 개발 프로세스 상의 보안 관리가 이루어져야 한다.

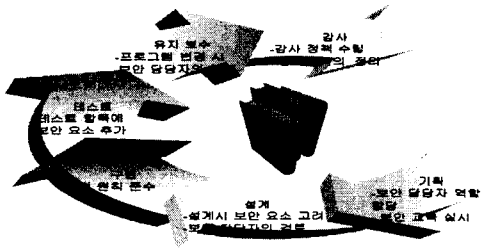

그림 2: 개발 프로세스 상의 보안 관리

1) 기획

웹 응용프로그램 개발 과정 전반에서 적절한 보안 수준을 확보하기 위해서는 보안 담당의 역할을 정의하고 할당하는 것이 중요하다. 보안 담당의 역할은 개발되는 응용프로그램의 특성이나 개발팀의 구성에 따라 다르게 설정될 수 있지만 일반적으로 다음과 같은 일을 수행해야 한다.

표 1: 보안 담당의 역할

단계	보안 담당의 역할
기획	보안 목표 파악 보안 교육 실시 보안 부분의 소요 자원 분석
설계	보안 모듈의 설계 설계안 검토
구현	구현물의 보안 요소 검토
테스트	보안 테스트 항목 정의 보안 테스트 방법 정의 보안 테스트 감독 및 지휘
운영 및 관리	웹 응용프로그램 변경시 보안 요소 검토

2) 설계

웹 응용프로그램에서 사용자 편의성(기능 포함)과 보안성은 일정 부분 대립되는 면이 존재하므로 설계시 양자를 적절히 조율해야 한다. 이를 위해 보안 요구 수준을 정확히 파악하여 설계 원칙을 정의한다. 아래와 같은 일반적인 설계 원칙을 바탕으로 적절한 설계 원칙을 정의하여 설계시 준수하도록 한다.

- 단순한 설계
- 입·출력 값의 적절한 검증
- 적절한 에러 처리
- 신뢰할 수 있는 컴포넌트의 재사용
- 확장성 확보
- 권한의 최소화 및 세분화

3) 구현

구현을 시작하기 전에 반드시 구현에 참가하는 사람들을 대상으로 보안상 유의사항과 설계 원칙

을 확인하는 과정을 가지도록 하며, 보안 담당을 통한 구현물의 보안 요소를 검토하는 과정을 거치도록 한다.

4) 테스트

테스트시 다음과 같은 과정을 통해 웹 응용프로그램의 보안 수준을 확인한다.

- 테스트 항목에 보안 관련 부분을 명시한다.
- 블랙박스 테스트뿐만 아니라 소스 및 설정 검토를 병행한다.
- 테스트 결과는 반드시 문서화한다.

5) 운영 및 관리

운영 및 관리시 웹 응용프로그램의 변경이 빈번이 일어날 수 있으므로 다음과 같은 사항을 고려한다.

- 운영전 반드시 설정환경 점검목록을 만들어 설정환경이 적절한지 확인한다.
- 주기적인 취약점 점검계획을 수립하여 실시한다.
- 웹 응용프로그램 변경시 보안 담당의 검토 후 변경 사항을 반영한다.
- 웹 응용프로그램의 운영 및 변경시 주요 사항은 반드시 문서화한다.

3. 신원 인증 및 접근제어

신원 인증은 사용자나 개체의 신원을 확인하는 과정을 말한다. 일반적인 사용자 인증은 ID와 비밀번호 확인을 통해 수행되고, 확인된 정보는 세션 관리를 통해 유지된다. 접근제어는 해당 사용자가 자원에 접근할 수 있도록 인가되었는지 확인하는 과정이다. 웹 응용프로그램에서 안전한 신원 인증 및 접근제어가 이루어지기 위해 다음과 같은 원칙이 필요하다.

- 명확한 신원 인증 및 접근제어 정책을 수립한 후 설계 및 구현에 반영한다.
- 신원 인증 및 접근제어 정보는 서버 측에서 관리한다.
- 최소한의 권한 부여 원칙을 준수한다.
- 신원 인증 및 접근제어 정책의 변경에 대비하여 확장성을 고려한다.

1) 신원 인증

웹 상의 신원 인증 방식은 크게 HTTP 기본 인증, HTTP 다이제스트, Form 기반의 인증, 인증서를 이용한 인증 등으로 나눌 수 있다. 요구 보안 수준에 따라 세 방식 중 적절한 것을 선택하여 사용하도록 한다.

- HTTP 기본 인증

'ID/비밀번호'를 통한 사용자 인증 방법을 사용하며 실패시 401 값을 리턴한다. 사용자 인증정보는 단순히 Base64 인코딩으로 전송된다.

- HTTP 다이제스트

'TTP 기본 인증' 방법을 개선하여 인증 정보의 해쉬값을 전송하는 방식이다.

- Form 기반 사용자 인증

HTML의 Form을 이용하여 사용자의 아이디와 비밀번호를 입력받아 인증을 수행하는 방식이다.

- 인증서를 이용한 인증

X.509 인증서와 같은 사용자 인증서를 이용하여 사용자의 신원을 확인하는 방법이다.

2) 접근제어

보통 신원 인증과 접근제어을 혼동해서 사용하는데, 실제로 신원 인증은 단순히 해당 개체가 진정 그 개체 맞는지 확인하는 과정이고 이 개체가 어떠한 자원에 접근할 권한이 있는지 확인하는 과정은 접근제어에 속한다. 만약 단순히 어떠한 그룹에 속하는 사용자가 그 그룹에서 다룰 수 있는 자원에 대해 읽고, 실행, 쓰기 및 삭제의 전 권한을 가지는 경우라면 신원 인증이 곧 접근제어가 될 수 있지만 이러한 경우는 극히 제한적인 환경으로 실세계에서는 훨씬 복잡한 보안 환경이 일반적이다. 웹 응용프로그램 개발에 앞서 다음 방식 중에서 요구되는 보안환경에 적합한 접근제어 방법을 선택한다.

- 강제적 접근제어

강제적 접근제어는 각 정보에 결합된 비밀등급(classification level)과 사용자에게 부여된 인가등급(clearance level)을 사전에 규정된 규칙과 비교하여 그 규칙을 만족하는 사용자에게만 접근 권한을 부여하는 보안정책이다.

- 임의적 접근제어

임의적 접근제어는 시스템내의 각 사용자(또는 사용자 그룹)와 각 객체에 대하여 사용자가 객체에 허용된 접근 모드를 기술하는 사용자의 식별(identification)과 인가에 기초하여 정보에 대한 사

용자의 접근을 제어한다.

■ 행위기반 접근제어

어떤 사용자가 업무흐름 내 단위업무에 대한 실행 권한을 부여받았다 하더라도 그 권한의 사용은 워크플로우의 진행 상태에 따라 제약을 받는 접근 제어 방법이다.

■ 역할기반 접근제어

역할기반 접근제어(Role-based Access Control)는 정보 자원에 대한 접근 권한을 직접 부여하지 않고, 역할을 통해 간접적으로 권한이 부여되는 방법이다. 이러한 방법은 권한 관리를 매우 단순화 시켜주고 기업의 특정한 보안정책을 구현하는데 있어서 유연성을 제공하는 장점이 있다. 사용자는 그들의 업무적 권한과 책임에 따라 특정 역할의 구성원이 되며 접근 구조의 변경이 없이도 역할을 쉽게 변경 가능하다.

4. 세션 관리

HTTP 프로토콜은 사용자 요청에 대한 서버 측 응답이 하나의 트랜잭션을 이루며 각각의 트랜잭션 별로 세션이 끊어진다. 이러한 특성으로 인해서 웹 응용프로그램 설계 및 구현시 안전한 세션 관리를 위한 고려가 필수적이다. 세션 관리 메커니즘이 안전하지 않은 경우 악의적인 사용자는 세션 하이재킹이나 세션 정보 위조 등을 통해 비인가된 기능을 수행할 수 있다. 안전한 세션 관리를 위해서 다음과 같은 원칙이 필요하다.

- 세션 관리 정책을 수립한 후 설계 및 구현한다.
- 세션 ID는 암호화를 통해 평문으로 전송되지 않도록 한다.
- 세션 ID의 유효 기간은 정상적인 서비스가 이루어질 수 있는 한에서 최소한의 시간을 선택한다.
- 세션의 생성 및 소멸에 대한 로그 관리를 수행한다.

1) 세션 관리 방법

세션은 세션 ID에 의해 관리되는데 이러한 세션 ID를 관리하는 방법에는 URL 기반 세션 관리, Hidden 필드를 이용한 세션 관리, 쿠키(Cookies)를 이용한 방법이 있다.[3] 최근에는 쿠키를 이용하는 방법이 주로 쓰이고 있으나, 각 방법마다 장단점이 있으므로 적절히 판단하여 방법을 상황에 적합한 방법을 선택한다.

■ URL 기반 세션 관리

세션 ID 정보가 HTTP GET 요청을 통하여 전달되는 형태로 URL에 세션정보가 추가된다. 브라우저의 쿠키 설정에 관계없이 사용 가능하고 단순히 URL을 복사해서 세션을 복사할 수 있으나 브라우저의 기록 등에 의해 세션 정보가 노출될 수 있다.

■ Hidden 필드를 이용한 세션 관리

특정 폼을 생성하고 세션 ID 정보를 폼 안의 Hidden 필드에 설정한 후 HTTP POST 방식으로 전송하여 세션을 관리하는 방법이다. 이 역시 쿠키 설정과 관계없이 사용할 수 있고, 'URL 기반 세션 관리'보다 안전하나 Hidden 필드가 많아지면 페이지가 복잡해지고 GET 방식을 이용할 경우 URL에 직접 세션 정보가 노출된다.

■ 쿠키 이용

쿠키는 세션 ID를 설정하는 방법으로 가장 널리 쓰이는 방법이다. 세션 ID의 유효 시간을 설정하여 관리할 수 있고, 대부분의 브라우저가 지원하므로 복잡한 코딩이 필요없지만 '영구 쿠키'는 시스템에 남아있기 때문에 오용될 소지가 있다.

■ 컴포넌트의 세션 관리 기능 이용

근래에는 CGI 엔진이나 미들웨어 등에서 세션 관리를 지원하는 경우가 있고, 실제로 이러한 기능을 이용하여 세션 관리를 수행하는 경우가 많다. 구현이 간편하지만 세션 ID의 길이 등 세부적인 설정에 제약이 많다.

1) 적절한 세션 ID값의 설정

웹 응용프로그램 세션 관리에서 가장 중요한 요소는 적절한 세션 ID 값을 설정하는 것이다. 세션 ID는 응용프로그램을 통해 인증된 사용자 정보를 추적하는데 사용되므로, 만약 악의적인 사용자가 이를 예측할 수 있다면 인가된 사용자로 위조하여 중요 정보에 접근할 수 있다. 따라서 세션 ID는 추측이 불가능하도록 임의의 값과 충분한 길이를 가져야 한다.

■ 세션 ID의 임의성

세션 ID는 통계적인 테스트를 통과해야 하며, 예측이 불가능하고 입력이 동일한 조건에서도 다른 값이 생성되어야 한다.

■ 세션 ID의 길이

악의적인 사용자가 세션 ID를 위조할 수 없도

록 충분히 긴 세션 ID를 사용해야 하지만, 전송 속도와 시스템 성능을 고려하여 적절한 길이로 설정한다.

5. 데이터 검증

웹 응용프로그램에서 발생하는 대다수의 보안문제는 데이터 검증을 통해 사전에 예방되거나 위협이 감소될 수 있다. 따라서 데이터 검증은 안전한 웹 응용프로그램을 설계하는 가장 중요한 요소이다. 여기에서 데이터 검증이란 함은 웹 응용프로그램의 입력과 출력 모두에 대한 검증을 의미한다.

데이터 검증 방식을 설계할 때 아래와 같은 세 가지 모델 중에서 적절한 방식을 선택한다.

- 유효한 데이터만 허용
- 알려진 침해 데이터 차단
- 침해 데이터 필터링

6. 이벤트 로깅

로깅은 웹 응용프로그램 및 관련 프로세스에 대한 보안 정보를 제공하므로 이벤트 시간, 시작 프로세스, 프로세스 소유자, 이벤트 설명과 같은 상세한 정보를 남긴다. 웹 서버와 응용프로그램의 로깅 성능을 소모하지 않기 위해 효과적인 로그 관리 및 수집 기능이 필요하고, 분리된 호스트에 암호화하여 로그를 남기고 주기적으로 백업한다.

7. 알려진 웹 취약점에 대한 대응

최근 들어 웹 응용프로그램에서 많은 취약점들이 발견되어 위협이 증가하고 있다. 예를 들어, Cross-Site Scripting(CSS) 취약점은 표 3처럼 폼 입력에 사용자 정보 유출이 가능한 악성 스크립트를 실행 가능케 한다.[4]

표 2: Cross-Site Scripting 취약점

• 정상적인 HTTP 요청
http://www.test.or.kr/test.cgi?userid=hong
• 악의적인 HTTP 요청
http://www.test.or.kr/test.cgi?userid=hong<script>document.write('<img src="http://targetsite.com'+document.cooke+'"</script>

이와 같이 알려진 취약점에 대응하기 위해 웹 취약점을 분류하고 취약점 별 예방대책을 수립한다. 알려진 취약점을 분류하면 다음과 같다.[5]

1) 사용자 피해 취약점

- Cross-Site Scripting

2) 시스템 피해 취약점

- SQL Injection
- 시스템 명령 실행
- 경로 탐색
- 널 바이트 문제
- 문자코드 변환
- URL 인코딩

3) 파라미터 조작

- 쿠키 조작
- HTTP 헤더 조작
- HTML 폼 필드 조작
- URL 조작

4) 기타

- 시스템 및 엔진 패치
- 시스템 설정 강화
- HTML 문서내 주석 제거
- 백업 및 임시 파일 삭제
- 디폴트 계정 삭제

예를 들어, CSS 취약점은 그림 XX처럼 폼 입력에 사용자 정보 유출이 가능한 악성 스크립트를 실행하는 것이다.

8. 프라이버시 보호

원격근무를 위한 웹 기반 시스템은 안전하지 않은 외부 PC에서의 접속에 대비하여 주민등록번호, 주소, 전화번호와 같이 민감한 개인정보가 유출되지 않도록 검토하는 절차가 필요하다. 개인 정보는 필수적인 페이지에서만 표시되어야 한다. 또한, 사용자의 브라우저 이력에서 민감한 정보가 남아 있지 않도록 모든 폼 입력은 POST 방식을 사용한다.

III. 결론

나날이 증가하는 웹 취약점에 의한 위협이 감소될 수 있다면 원격근무를 위한 웹 기반 업무 시스템이 보다 안전하게 서비스될 수 있으며 업무 효율이 증가할 것이다. 이를 위해 개발 단계 전반에서 보안 요소를 고려하여 안전한 업무 시스템을 구축해야 한다.

본 논문에서는 요구 수준에 부합하는 보안 목표를 설정하여, 전체 개발 프로세스 상의 보안 관리 활동을 기술하였다. 또한, 신원 인증 및 접근제어, 세션 관리, 이벤트 로깅, 데이터 검증, 알려진 취약점 대응, 그리고 프라이버시 보호와 같이 시스템의 안전한 구축을 위한 보안 요소들을 분석하였다.

향후 과제로는, 원격근무를 위한 웹 기반 업무 시스템 구축시 분석된 보안 요소를 적용하는 연구를 수행할 것이다.

참고문헌

[1] 서정석, 김한성, 조상현, 차성덕, "웹 어플리케이션 특성 분석을 통한 공격 분류", 정보과학회논문지: 정보통신 제30권 제1호, pp. 95-116, 2003년 2월.

[2] 서정택, 정윤정, 임을규, 김인중, 이철원, "인트라넷 취약점 분석·평가 방법론 연구", 한국정보과학회 2003봄 학술발표논문집(A), pp. 296-298, 2003년 4월.

[3] Gunter Ollmann, "Web Based Session Management", http://www.technicalinfo.net

[4] 김형주, 예홍진, 조은선, "접근 제어를 이용한 교차 사이트 스크립트 필터링", 한국정보과학회 2002봄 학술발표논문집(A), pp. 799-801, 2002년 4월.

[5] Mark Curphey, David Endler, William Hau, Steve Taylor, Tim Smith, Alex Russell, Gene McKenna, Richard Parke, Kevin McLaughlin, "A Guide to Building Secure web Applications", http://www.owasp.org

[5] Gunter Ollmann, "Web Based Session Management", http://www.technicalinfo.net