SigPL Secure Coding Workshop

웹 어플리케이션 취약점 대응을 위한 OWASP 시큐어 코딩 소개

OWASP Korea Chapter
조민재 이사
johnny.cho@owasp.org

OWASP
2013. 06. 28

The OWASP Korea Chapter
http://www.owasp.or.kr

발표자 소개

- OWASP Korea 챕터 이사(Board Member)
- Yahoo! Korea, NHN, 펜타시큐리티 등 기업체 경력 13년
- SW개발 보안 및 인프라 보안 전문가
- OWASP Top10 2010 번역 참여

발표 내용

1. OWASP 소개
2. OWASP 시큐어 코딩 규칙 소개
3. Q&A

OWASP

OWASP (www.owasp.org)

- The Open Web Application Security Project
- 응용 소프트웨어 보안을 연구하는 비영리 단체
- 2001.12 : 온라인 조직으로 미국에서 처음 시작
- 2004.04 : 정식 비영리 법인 출범
- 웹 소프트웨어 보안에 관련된 연구 프로젝트 수행
  - 시험 도구 및 표준 문서 발간
  - 서적 발간 : 시험, 안전한 코드 개발, 보안 코드 검토
  - 표준 보안 통제와 라이브러리
  - 전 세계 챕터(지부) 운영 및 교류
  - 보안 컨퍼런스(AppSec) 개최

OWASP

- **핵심가치**
  - OPEN(개방) : 모든 것이 투명하게
  - INNOVATION(혁신) : 혁신과 경험을 존중
  - GLOBAL(세계적) : 누구든지 참여
  - INTEGRITY(진실) : 진실되고, 성실하게
- **운영원칙**
  - Free & Open
  - Governed by rough consensus & running code
  - Abide by a code of ethics
  - Not-for-profit
  - Not driven by commercial interests
  - Risk based approach
- **미션**
  - 응용 소프트웨어 보안성 향상을 목표로 활동
  - 상용 제품과 서비스에 대한 지지하거나 추천하기 않고, 대신 다양한 지식을 모아서 응용 소프트웨어 보안기술을 생성

OWASP Korea Chapter

- 2011.01 : 온라인으로 이사진 구성 시작
- 2011.06 : 시큐어 코딩 규칙 참고 가이드 번역
- 2011.10 : 제 1회 OWASP Korea 챕터 세미나

- 2013.02 : OWASP AppSec ASIAPAC 2013
  - 2013.02.19~23, 하얏트 리젠시 제주

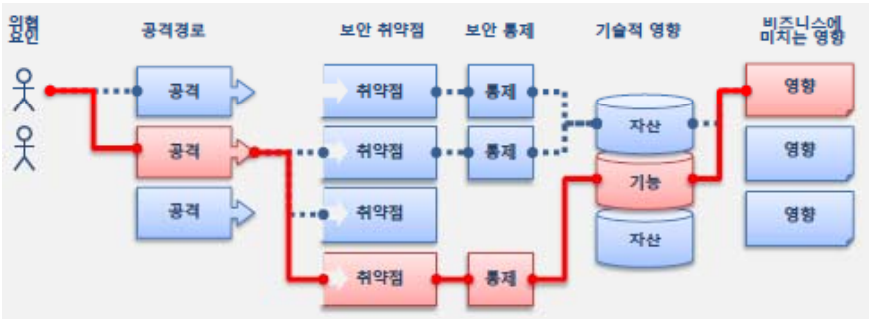
OWASP Top Ten (2013 Edition)

https://owasp.org/index.php/Category:OWASP_Top_Ten_Project

OWASP
The Open Web Application Security Project
http://www.owasp.org

OWASP Top10

http://owasp.org/index.php/Category:OWASP_Top_Ten_Project



위협요인	공격경로	취약점의 알려진 정도	취약점의 탐지 용이도	기술적 영향	비즈니스에 미치는 영향
?	쉬움	널리 알려짐	쉬움	심각	?
	보통	보통	보통	보통	
	어려움	일반적이지 않음.	어려움	미미	

OWASP 9

OWASP 시큐어 코딩 규칙

OWASP 10

문서에 대한 배경

- OWASP Secure Coding Practices – Quick Reference Guide
- 웹 어플리케이션 개발자들이 "**시큐어 코딩 규칙**"을 빨리 이해할 수 있도록 개발
- 보잉社 내부에서 사용하기 위해 최초 개발
- 2010.07 : 보잉사가 OWASP에 저작권을 양도
- 2010.08 : OWASP 프로젝트로 시작 (v1)
- 2010.10 : v2 개정

시큐어 코딩 규칙 개요

- 기술에 종속적이지 않은 코딩 규칙
- **무엇**을 할 것인가에 초점(어떻게 할 것인가는 아님)
- 집약적이면서도 포괄적인 **체크리스트** 형식
- 취약점보다는 **시큐어 코딩 요구사항**에 중심
- 개발자와 보안담당자간에 동일한 용어로 의사소통하기 위한 **기술 용어** 정의

구성

- 체크리스트 형식으로 되어 있으며, 아래 내용이 포함
  - 목차
  - 소개
  - 소프트웨어 보안과 위험 원칙 개요
  - **시큐어 코딩 규칙 체크리스트**
  - 참고 자료
  - 중요한 기술 용어

체크리스트 규칙

- 간단하고, 핵심 위주
- "해야 할 것"과 "하지 말아야 할 것"을 직설적으로 표현
- 규칙을 서열화 하지 말자
- 일부 규칙은 시스템 또는 정보의 중요도에 따라 조건적으로 권고
- 어플리케이션에 적용된 보안 사항은 확실히 이해할 수 있어야 한다.

시큐어 코딩 규칙 체크리스트

- 입력 값 검증
- 출력값 인코딩
- 인증과 패스워드 관리
- 세션 관리
- 접근 통제
- 암호 규칙
- 에러 처리와 감사기록
- 데이터 보호
- 통신 보안
- 시스템 설정
- 데이터베이스 보안
- 파일 관리
- 메모리 관리
- 일반 코딩 규칙

체크리스트 부분

■ 입력 값 검증
- 신뢰 시스템의 모든 데이터 유효성을 검증 (예: 서버)
- 모든 데이터의 출처를 확인하고 신뢰/비신뢰로 구분하라. 출처를 신뢰하지 못하는 모든 데이터를 검증 (예: 데이터베이스, 파일 스트림 등)
- 잠재적으로 위험한 문자가 반드시 입력되어야 한다면, 출력 값 인코딩, 안전한 작업을 위한 API, 그리고 애플리케이션 전체에 대한 데이터가 활용되는 감사 기능과 같은 추가적인 통제기능 구현. 일반적으로 위험한 문자의 예는 다음과 같다:
  - < > " ' % ( ) & + ₩ ₩' ₩"

체크리스트 부분

■ 출력 값 인코딩
- 신뢰 시스템의 모든 인코딩을 적용(예: 서버)
- 외부로 나가는 인코딩에 대해 표준적이고, 시험이 된 루틴 이용
- 애플리케이션의신뢰범위*(trust boundary) 외부로 부터 클라이언트로 반환된 모든 데이터는 상황에 맞는 출력 값 인코딩. HTML 엔터티 인코딩 이 예가 될 수 있지만, 모든 경우에 그런 것은 아니다.*
- 사용하는 인터프리터에서 안전하다고 알려진 문자 이외의 모든 문자를 인코딩
- SQL, XML, 그리고 LDAP에 쿼리하는 신뢰받지 못하는 데이터의 모든 출력 값을 상황에 맞게 필터링

체크리스트 부분

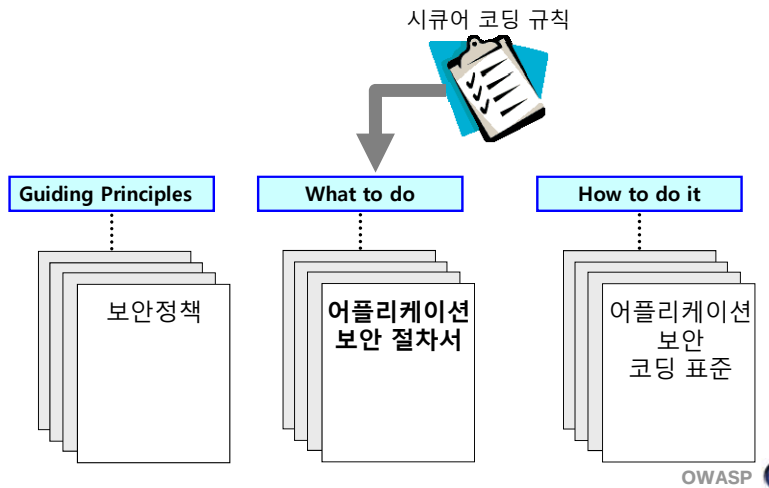
■ 인증과 패스워드 관리
- 의도적으로 공개하는 경우를 제외한 모든 페이지와 자원은 인증 실시
- 모든 인증 통제는 반드시 신뢰 시스템에서 수행(예: 서버)
- 특수한 경우를 제외하곤, 표준화되고 검증된 인증 서비스를 확립하고 이용
- 외부 인증 서비스를 호출하는 라이브러리를 포함한 모든 인증 통제는 중앙 집중화하여 사용하라
- 요청된 자원과 인증 로직을 구분하고 중앙 집중화된 인증 통제 정보를 주고받는 데는 리다이렉트를 사용하라
- 모든 인증 통제는 안전하게 인증 실패 처리
- 인증 정보를 전송할 때에는 반드시 HTTP POST 요청을 사용
  - GET method : 정보가 URL에 적혀서 전달
  - POST method : 정보가 request 본문에 전달됨

데이터베이스 보안

- ❑ 반드시 사전 준비된 쿼리(prepared queries) 를 사용
- ❑ 입력 값 검증과 출력 인코딩을 활용하고 **메타 문자(*, ?, [ ],~, !, +) 를 확실히 처리**. 만약 이 작업이 실패한다면, 데이터베이스 명령어를 실행시켜선 안됨
- ❑ 반드시 변수가 제대로 입력되었는지 검증
- ❑ 데이터베이스에 접속했을 때 애플리케이션은 가장 낮은 수준의 권한을 사용
- ❑ 데이터베이스 접속을 위해 안전한 인증정보를 사용
- ❑ 접속 정보는 애플리케이션 내에 하드코딩 되어선 안됨. 접속 정보는 신뢰된 시스템의 별도 설정 파일로 저장 및 암호화
- ❑ 데이터에 직접 접근하지 못하도록 저장 프로시저(stored procedure)를 사용하고, 데이터베이스에 존재하는 기본 테이블에 대한 권한 제거
- ❑ 데이터베이스 접속은 가능한 빨리 종료
- ❑ 기본으로 제공되는 데이터베이스 관리 계정을 제거하거나 패스워드를 변경. 강력한 패스워드/비밀문구를 활용하거나 다중 인증을 구현
- ❑ 불필요한 모든 데이터베이스 기능을 비활성화(예: 불필요한 저장 프로시저 또는 서비스, 유틸리티 패키지, 필요한 최소한의 기능과 옵션만을 설치하라(공격 노출 범위 축소))
- ❑ 불필요한 벤더 기본 콘텐츠 제거(예: 샘플 스키마)
- ❑ 비즈니스 요구사항에 필요하지 않은 모든 기본 계정 비활성화
- ❑ 애플리케이션은 데이터베이스에 접속할 때 개별 기능(권한)에 대해 다른 인증정보를 사용 (예: 사용자, 읽기 전용 사용자, 방문자, 관리자)

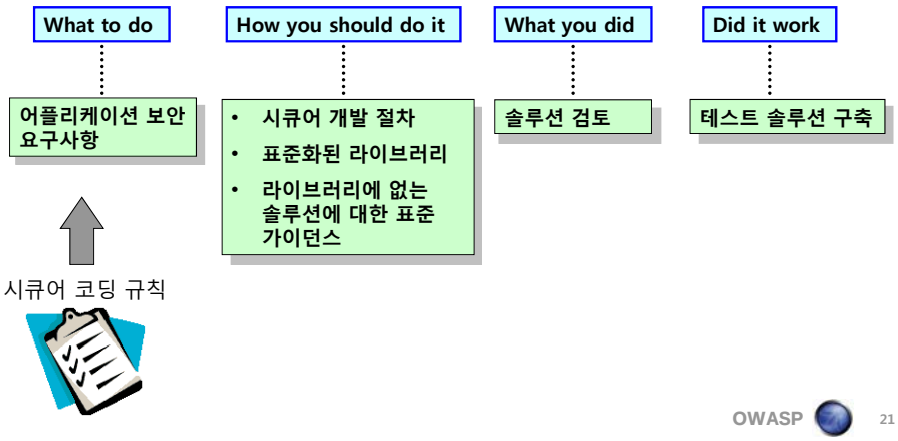
활용 방법

■ 시나리오 #1: 가이던스 문서 개발



활용 방법 *continued*

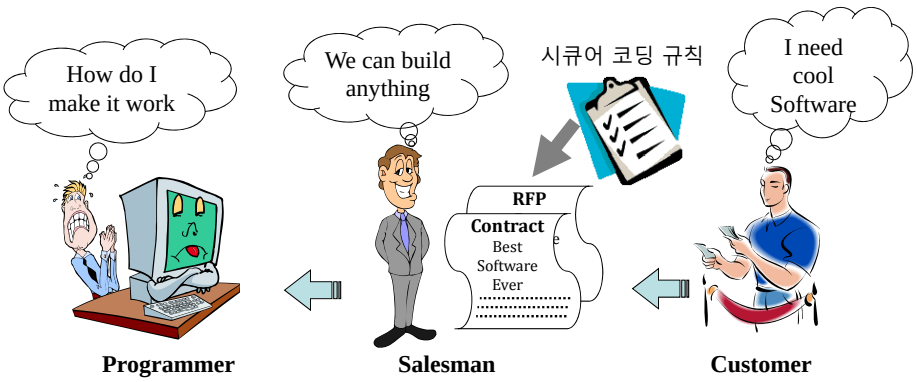
■ 시나리오 #2: SDLC(Secure Development Lifecycle) 지원



가이드 활용 방법 *continued*

■ 시나리오 #3: 개발 위탁

- 소프트웨어 외주 개발을 할 때 보안요구사항을 확인
- RFP 및 계약서에 포함 (가격에도 꼭 포함!)



안전한 개발 프레임워크

안전한 소프트웨어 개발 프레임워크 가이드는 시큐어 코딩 규칙범위는 아니지만 OWASP 프로젝트에 다양한 프로젝트가 있음

- Implement a secure software development lifecycle
  - OWASP CLASP Project
- Establish secure coding standards
  - OWASP Development Guide Project
- Build a re-usable object library
  - OWASP Enterprise Security API (ESAPI) Project
- Verify the effectiveness of security controls
  - OWASP Application Security Verification Standard (ASVS) Project)
- Establish secure outsourced development practices including defining security requirements and verification methodologies in both the RFP and contract
  - OWASP Legal Project

Q & A
Suggestion