외주 개발 웹 어플리케이션 테스팅의 보안성 강화 방안

최경호* · 이동휘**

요 약

웹 서비스를 가능하게 하는 웹 어플리케이션은 내부 개발자가 보안 의식을 갖고 만든다면 일정 수준 이상의 안전성을 보여준다. 하지만 외주 개발의 경우, 품질의 우수성보다는 요구 사항을 충족하고 요청 받은 기능을 실행시키는데 주안점이 있기 때문에 안전성이 우선되지 못한다. 따라서, 본 논문에서는 소프트웨어에 대한 객관적이고도 독립적인 시각으로의 평가를 가능하게 해주는 소프트웨어 테스트 절차를 보안 중심으로 개선하였다. 제안된 모델은 웹 어플리케이션의 외주 개발 시에도 초기부터 보안을 고려할 수 있게 해주며, 특히 보안 인식이 부족한 상황에서 작성된 프로그램의 수정 소요 발생으로 인한 개발 일정 지연 사태를 미연에 방지할 수 있는 효과가 있음을 확인하였다. 이러한 결과는 자원관리체계를 중심으로 웹 어플리케이션에 대한 소요가 증가하고 있는 국방 분야에서도 엄격한 테스트를 토대로 보안 취약점을 지닌 채 서비스되는 것을 방지할 수 있기에 활용이 가능할 것으로 판단된다.

Enhanced Security Measurement of Web Application Testing by Outsourcing

Kyong-Ho Choi* · DongHwi Lee**

ABSTRACT

A web application that allows a web service created by a internal developer who has security awareness show certain level of security. However, in the case of development by outsourcing, it is inevitable to implement the development centered on requested function rather than the issue of security. Thus in this paper, we improve the software testing process focusing on security for exclusion the leakage of important information and using an unauthorized service that results from the use of the vulnerable web application. The proposed model is able to consider security in the initial stage of development even when outsourced web application, especially, It can prevent the development schedule delay caused by the occurrence of modification for program created by programer who has low security awareness. This result shows that this model can be applied to the national defense area for increasing demand web application centered resource management system to be able to prevent service of web application with security vulnerability based on high test.

Key words : Software Testing, Security Test, Security Architecture, Security Management, Application Security

접수일(2015년 5월 4일), 수정일(1차: 2015년 5월 17일), 게재확정일(2015년 5월 28일)

* (주)제이에프테크 연구소장
** 동신대학교 정보보안학과 조교수(교신저자)

1. 서 론

인류가 정보화 사회로 진입한 이후부터는 많은 서비스가 웹을 통해 제공되고 있다. 또한 웹 서비스는 직접적으로 경제적 부가가치를 창출할 뿐만 아니라 브랜드 홍보, 대고객과의 접점 및 대내적인 관리 창구로 활용되는 등 기업과 조직의 성장을 위한 유·무형적인 기여도가 크기에 그 활용도가 급격히 증가하고 있다. 국방 분야 또한 자원관리체계를 중심으로, 대내외 서비스의 제공 그리고 관리, 운용적 측면에서의 웹 서비스 활용이 증가하고 있다.

그런데 웹을 이용한 서비스들은 안전한 환경을 필요로 한다. 웹 서비스 제공자가 의도한 대로 정보의 저장과 전달이 이루어져야지만, 사용자들로부터 좋은 평가와 계속적인 이용을 통한 고부가가치를 이끌어 낼 수 있기 때문이다. 그래서 웹 서비스를 제공하는 어플리케이션들의 취약점에 대비하고 신뢰성을 향상시키기 위해 가상화 기술을 적용하는 방법 등과 같은 비인가된 사용자의 변조, 무단 사용 및 정보 절취 등의 악성 행위로부터 보호하려는 노력들이 계속되고 있다[1].

웹 어플리케이션은 내부 개발자가 보안 의식을 갖고 만든다면 일정 수준 이상의 안전성을 보여준다. 하지만 외주 개발의 경우, 품질의 우수성보다는 요구 사항을 충족하고 요청 받은 기능을 실행시키는데 주안점이 있기 때문에 안전성이 우선되지 못한다. 게다가, 짧게 주어지는 어플리케이션 개발 일정이란 어려운 환경 속에서는 더욱 더 보안에 대한 신경을 쓰기가 어렵다. 결국, 안전하지 못한 외주 개발 웹 어플리케이션으로 인해, 내부의 정보가 유출될 소지가 있게 되는 것이다. 따라서, 웹 어플리케이션의 외주 개발 시, 보안성을 높이기 위한 노력들이 함께 이루어져야 하는 것이다.

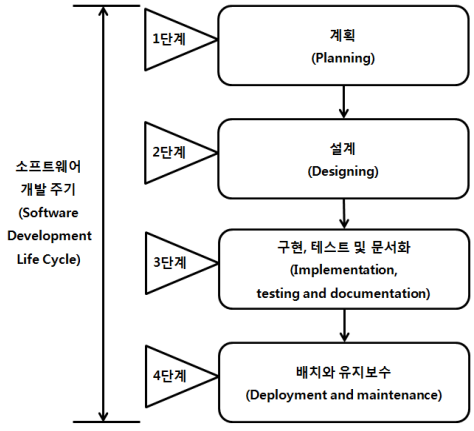
본 논문에서는 외주 개발 웹 어플리케이션의 보안성을 높이기 위해 개선된 테스팅 모델을 제안한다. 이 모델은 개발 계획 수립 단계부터 유지보수에 이르기까지 외주 개발 웹 어플리케이션의 전 단계에 걸친 보안성 강화 방법을 제시하여, 관리자로 하여금 보다 신뢰성 있는 웹 어플리케이션을 획득할 수 있게 해준다.

이어지는 2장에서는 관련 연구로 웹 어플리케이션 개발 시 적용할 수 있는 소프트웨어 개발 주기를 보안 관점에서 살펴보고, 이 관점으로 외주개발 웹 어플리케이션을 평가하는 테스팅을 검토해본다. 3장에서는 외주 개발 웹 어플리케이션의 테스팅 방법을 개선하여 보안성을 높일 수 있는 모델을 제시하고, 4장에서 이를 평가한다. 마지막으로는 본 연구의 한계와 다음 연구 방향을 고찰해 보기로 한다.

2. 관련 연구

2.1 안전한 소프트웨어 개발 주기

웹 어플리케이션을 비롯한 모든 소프트웨어의 의 개발 및 관리는 일반적으로 다음 [그림 1]에서 볼 수 있는 바와 같이, 계획, 설계, 구현, 테스트 및 문서화 그리고 배치와 유지보수, 이렇게 4단계로 이루어진 소프트웨어 개발 주기를 따른다.



[그림 1] 소프트웨어 개발 주기

그리고 소프트웨어 취약점으로부터의 발생 가능한 위협을 방지하기 위해서는 소프트웨어 개발 주기 모든 단계에 걸쳐 보안 거버넌스(Security Governance)가 고려되어야 한다[2]. 이러한 관점은 국내외 여러 기업이나 기관들에서 자체적으로 소프트웨어 개발 보안 절차를 개발하여 활용하는데 적용되고 있다[3].

그러나 보안 중심의 소프트웨어 개발 주기는 웹 어

플리케이션을 외주 개발하는 경우에 적용하기 어렵다. 소프트웨어의 개발 주체가 계약 관계에 있는 타 업체 또는 인력이기 때문에 내부적으로 수립하여 운영하고 있는 규정과 지침을 강제할 수 없다. 그리고 보안 중심의 소프트웨어 개발 주기에 외주 개발 주체를 동참시키더라도, 시간과 장소의 제약 때문에 올바르게 준수되고 있는지를 감독하기가 힘들다.

그래서, 웹 어플리케이션을 외주 개발하는 경우, 해당 서비스의 소유자 또는 사용자가 직접 테스트를 수행하여 기능과 보안성에 대한 검증을 진행한다.

2.2 소프트웨어 테스팅

소프트웨어 테스팅은 개발된 소프트웨어에 대한 객관적이고도 독립적인 시각으로의 평가를 가능하게 해준다. 이를 통해 비정상적인 프로그램 동작, 오류 발생 등의 문제를 찾아낼 수 있기 때문에, 웹 어플리케이션의 기능과 보안성 개선에 좋은 방법으로 활용될 수 있다.

소프트웨어 테스팅의 V-Model은 소프트웨어 개발 주기에 맞춘 테스팅을 진행할 수 있게 한다[4]. 이 모델은 진행하는 소프트웨어 개발 주기의 모든 단계에서 각 단계별로 검증을 완료해야 다음 단계를 시작할 수 있게 한다. 그러나 아직은 기능 분야에 치중하고 있고, 보안은 하나의 부분으로 시행되거나 병행하여 진행되는 정도이기에 보안 취약점의 발견 시 수정 및 보완이 어렵다. 국방 분야에서도 기능 중심의 소프트웨어 개발 및 테스팅은 세세하게 이루어지고 있으나[5], 보안 테스트는 기능과 별개로 이루어지거나 사후적으로 추진되고 있다.

그리고 웹 어플리케이션을 외주 개발하는 경우, 시험, 평가를 실시하는 테스트 단계 외에는 발주처가 관계하기 어렵다. 이러한 상황은 개발 주체와 보안 담당 부서의 긴밀한 협조 없이 소프트웨어를 개발하는 경우에도 마찬가지이다. 따라서, 테스트 단계에서는 미리 정립된 평가 범위와 항목들을 토대로 하여, 요구사항 대로 설계 및 구현되었는지, 보안 취약 요소는 없는지, 기능은 올바르게 동작하는지 등을 점검하고, 예상 가능한 문제점들을 제거할 수 있도록 진행하여야 할 것이다. 그리고 직접적으로 관여할 수 없는 테스트 분야 외의 소프트웨어 개발 주기에 관해서는 보안 가이드를 제공하거나 테스트 방법을 확장하여 사후 검증을 하는 방식 등으로 접근하여 외주 개발 주체의 개발 과정을 간접적으로라도 관리하여 보안성을 높이도록 하여야 할 것이다.

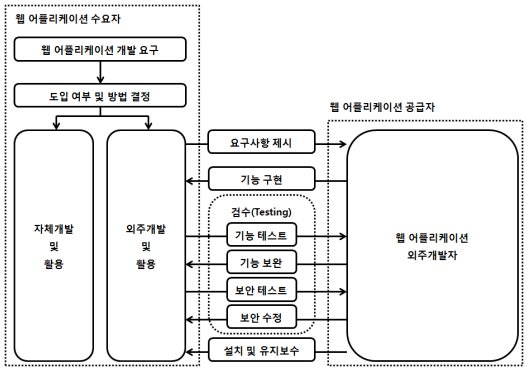
3. 보안성이 강화된 외주 개발 웹 어플리케이션 테스팅 모델

웹 어플리케이션을 직접 활용해야하는 주체가 개발까지 수행한다면, 소프트웨어 개발 주기 모든 단계에 보안 거버넌스를 고려할 수 있고, V-Model을 활용한 각 단계별 이행 상황 평가도 가능하다. 그러나, 개발 능력의 부재, 관리 상의 효율성 또는 경영 상의 이유 등으로 인해 웹 어플리케이션이 외주 개발되는 경우도 많다.

3.1 기존 외주 개발 웹 어플리케이션 테스팅 절차

제기된 웹 어플리케이션의 소요가 어떠한 이유로든지 외주 개발로 결정되어 추진된다면, 발주처에서는 테스팅 절차가 도래할 때까지 해당 웹 어플리케이션의 실체를 접하기가 어렵게 된다. 웹 어플리케이션의 구조나 동작 방식 등의 설계, 코드 작성 및 서버 환경 구성 등 일련의 작업들이 외부의 주체에 의해 이루어지기 때문에 쉽게 접근할만한 여건이 안 되는 것이다. 또한, 외주 개발이란 방법으로 관리적 부담을 줄이는 것이기 때문에 관심 또한 적어지게 된다. 이로 인해 발주처에서는 테스팅 절차에 이르러서야 외주 개발된 웹 어플리케이션의 실체를 보게 되는 경우가 잦다.

현행 웹 어플리케이션의 테스팅 절차는 [그림 2]와 같이 기능 분야 점검 후 보안을 검토하는 형태를 취하고 있다. 이는 서비스를 위한 기능이 우선시되고, 보안은 사후적으로라도 수정 및 보완이 가능한 것이라는 관점을 반영한다. 안전보다는 서비스의 제공이 더욱 더 중요하다는 것이다.

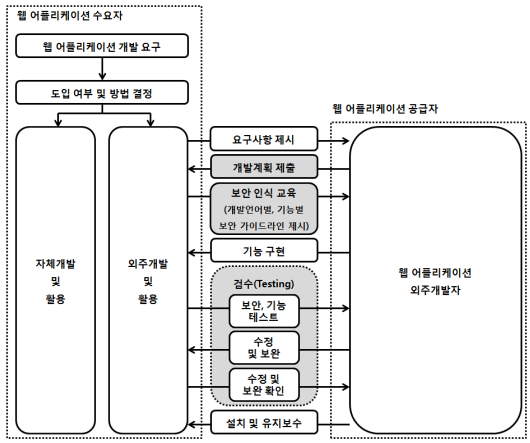


[그림2] 기존 외주개발
웹 어플리케이션 테스팅 절차

기존 웹 어플리케이션 테스팅 절차 하에서는 개발 일정과 서비스의 중요도에 따라 보안 수준의 편차가 발생할 소지가 있다. 만약, 보안 테스트 기준이 간과된 채 일반적인 요구 사항만을 충족하여 개발되거나 기능 구현에 치중하여 보안이 고려되지 않은 경우와 같은 재개발 소요가 발생하면, 서비스 제공과 보안 취약점 제거 중에서 선행되어야 할 것을 결정하게 되는 것이다.

3.2 보안성이 강화된 외주 개발 웹 어플리케이션 테스팅 절차

외주 개발 시에는 개발 주체가 발주처의 관리 범위 밖에 있기 때문에, 개발 단계별 어떠한 상황이 진행되고 있는지 알 수 없다. 또한 외주 개발 시 결과가 품질 측면에서 만족스럽지 못한 경우가 예전부터 있어 왔기 때문에[6], 능력이 우수한 개발 주체를 선정하는 것 뿐만 아니라, 개발 과정을 관리하여야 할 필요성이 제기되고 있는 것이다. 더욱이 보안 인력이 부족하고, 보안 인식이 낮은 현 소프트웨어 개발 환경에서는 외주개발 주체가 보안 분야의 전문성을 갖추기가 어렵기 때문에, 외주 개발 웹 어플리케이션의 보안성을 높이기 위해서는 테스팅 절차를 강화하여, 예상되는 위협원들을 제거하기 위한 노력을 수행해야 할 것이다. 이러한 필요성을 기반으로 본 연구에서 제안하는 보안성이 강화된 외주 개발 웹 어플리케이션 테스팅 절차는 다음 [그림 3]과 같다.



[그림3] 보안성이 강화된 외주개발
웹 어플리케이션 테스팅 절차

제안하는 모델은 외주 개발 주체가 작업을 시작하기 전에 사용하려는 언어와 모듈, 프로그램 구조 설계 결과 및 데이터베이스 정의서 등의 제출을 요구하여, 이를 검토한 후 개발 언어별 시큐어코딩 가이드, 기능별 취약점 정보 및 공격 사례 등을 제공한다. 코딩 규칙의 준수는 보안약점 감소 효과를 가져다주며, 기 공개된 취약점 정보, 유사 제품을 대상으로 한 공격 사례를 분석하는 것은 보안 테스팅 수행에 도움이 되기 때문이다[7, 8]

그리고 기존과는 다르게 기능과 보안을 동일한 단계에서 테스트한다. 기존에는 기능 부분의 테스팅을 통과하지 못하면 보안 부분에서는 전혀 점검을 할 수 없기 때문에, 기능 수정 이후 보안을 점검하여 다시금 기능 구현 방법을 수정해야 하는 경우도 발생한다. 이렇게 기능과 보안의 테스팅 관점이 분리되어 있는 것은 외주 개발 웹 어플리케이션의 테스팅 기간이 길게 지연되게 하는 이유가 된다. 그래서 제안된 모델에서는 기능과 보안 분야의 테스팅을 개별적으로 진행하긴 하되, 수정 요구 사항을 통합하여 산출함으로써 외주 개발 주체의 수정 작업 방향을 더욱 더 정확하게 제시해 줄 수 있다. 단 개선된 모델의 경우, 어느 한 분야가 점검 종료되었다고 해서 소스코드를 수정할 수 있는 것은 아니다. 소스코드의 수정은 다른 분야의 점검 중인 상황을 방해할 수 있기 때문에, 두 분야 모두 점검이 종료된 것을 확인한 이후에 보완을 진행해야 한다.

4. 제안 모델 평가

본 논문에서 제안하는 모델은 소프트웨어 테스팅을 전문적으로 수행하는 I기업에서, 보안이 중요한 관심사로 대두되는 상황 하 테스팅 방법론을 발전시키는 과정에서 제시되었다.

테스트 항목은 기능과 보안 분야의 담당자들이 각각 별도로 정의하여 작성하였다. 기능은 케이스를 기준으로 결함을 찾는 방법을 사용하였고, 보안은 일반적으로 알려진 자동화된 점검도구를 활용하는 방법[8] 외에도 최신 해킹공격의 재현 및 법/제도적 필수 요구사항의 구현 여부를 점검하는 방법을 사용하였다.

그리고 기존 모델은 D지사에서 기능과 보안을 단계별로 점검, 제안 모델은 S지사에서 기능과 보안을 통합하여 점검하는 방법으로 적용하여, 테스트를 받아야 하는 외주개발 주체를 다르게 함으로써, 평가 진행 간 서로 영향이 없도록 하였다.

여기서 보안 테스트는 총 35항목으로 다음 <표 1>에서 볼 수 있는 바와 같이 OWASP Top 10 2013[1]에서 볼 수 있는 위협원에 대한 대응방법의 적용, 기술적, 관리적 보호조치의 이행 및 개인정보보호법에 명시된 사항의 충족 등 안전한 웹 서비스 제공을 위해 필수적으로 갖추어야 할 사항들을 토대로 구성되었다.

<표 1> 보안 테스트 항목 예

순번	항목	점검방법	점검결과
1	개인정보보호정책 고지	홈페이지 확인	통과
2	업로드 취약점	확장자 변경	해당 없음
3	비인가 정보 열람	URL 직접접근	보완 필요
4	백업 또는 테스트 파일 노출	자동화 점검 도구	삭제 필요 URL List
5	중요 정보 암호화	패킷 모니터링	통과
6	패스워드 생성 및 관리	패스워드 생성 및 정책 확인	특수문자 사용 권고

평가는 D지사와 S지사에서 각각 5개씩의 프로젝트를 대상으로 진행되었으며, 다음 <표 2>에서 볼 수 있는 바와 같이, 제안된 모델을 이용한 테스트가 더 짧은 기간이 소요되고, 발견된 결함의 수도 적었음을 확인할 수 있다. 이는 보안 테스트 내용을 사전에 인지하고 웹 어플리케이션 개발을 하였기에, 작게는 결함의 발견 숫자가 적었고, 크게는 보안 취약점 제거를 위한 프로그램 구조 변경이 발생하지 않아 수정 및 보완 기간이 짧았기 때문이다.

<표 2> 제안 모델 평가 결과

구분	테스트 소요 기간(일)	수정, 보완 차수	평균 보안결함 수
기존 모델	7.8	3	12
제안 모델	5.4	2.2	6.4

제안 모델을 평가하는 과정에서 산출된 보안 테스트 결과의 주요 내용들을 살펴보면, 업로드 취약점 항목에서는 클라이언트 단에서 업로드 파일의 확장자를 검사하기 때문에 Paros와 같은 도구를 이용한 파일 전송 단계에서의 파일 바꿔치기가 가능한 경우, 게시글에 파일을 첨부하여 업로드 할 때는 확장자를 검사하나 게시글을 수정하는 과정에서는 확장자를 검사하지 않아 기존 첨부 파일을 실행 가능한 파일(예 : exe)로 바꾸는 것이 가능한 경우 그리고 보안 인식이 낮은 개발자가 게시판을 생성하여 첨부 파일의 확장자

1) 한국어 버전 위키백과에서는 OWASP(The Open Web Application Security Project)에 대해 웹에 관한 정보노출, 악성 파일 및 스크립트, 보안 취약점 등을 연구하여, 10대 웹 애플리케이션 취약점(OWASP TOP 10)을 발표하는 프로젝트로 설명하고 있다. OWASP TOP 10은 웹 애플리케이션 취약점 중에서 빈도가 많이 발생하고, 보안상 영향을 크게 줄 수 있는 것들 10가지를 선정하여 2004년, 2007년, 2010년을 기준으로 발표되었고, 문서가 공개되었다. OWASP는 3년 주기로 문서를 발표하고 있으며, 가장 최신본은 2013년 버전이다.

를 아예 검사하지 않는 경우들이 수정해야 할 취약점으로 지적되었다.

자동화 도구를 이용할 때에는 검사하고자 하는 도메인에서 서비스 하는 모든 URL들을 무작위 대입하여 찾아내기 때문에, 비인가 정보 열람 항목에서 점검하는 인증 없이 직접 접근이 가능한 URL들을 찾아낼 수 있었다. 이 URL들은 실제로 인증을 점검하지 않는 URL들인 경우도 있었고, 좀 더 다양하고 시각적으로 보기 좋은 서비스들을 제공하기 위해 사용한 IFRAME들인 경우도 있었다. IFRAME의 경우는 부분적인 메뉴나 정보들 일지라도 비인가자들에게 접근을 허용하면 중요 정보 조합의 단서로 사용될 수 있기 때문에 인증된 사용자만이 열람 가능해야 한다. 그리고 백업 또는 테스트 파일 노출 항목에서 점검하는 개발자가 남겨둔 임시 파일들이나 비인가자들이 열람하지 못해야 하는 백업 파일들을 찾아내어 삭제를 권고할 수 있게 하였다.

5. 결 론

본 논문에서는 보안에 취약한 외주개발 웹 어플리케이션 사용으로 인한 내부 정보의 유출 및 비인가된 서비스 사용을 차단하는 등 보안성을 높이기 위하여, 소프트웨어 테스트 절차를 보안 중심으로 개선하였다. 제안된 모델은 웹 어플리케이션의 외주개발 시에도 초기부터 보안을 고려할 수 있게 해주며, 특히 보안 인식이 부족한 상황에서 작성된 프로그램의 수정 소요 발생으로 인한 개발 일정 지연 사태를 미연에 방지할 수 있는 효과가 있음을 확인하였다.

그리고 본 논문에서 제안된 보안성이 강화된 외주개발 웹 어플리케이션 테스팅 모델은 기존의 기능과 보안 테스트 각각을 분리하여 운영하던 것을 통합하여, 중요도나 우선 순위를 가리지 않고 동일한 관점에서 발견되는 문제점을 해결할 수 있게 하였다. 이러한 점은 보안취약점이 내재된 채 서비스를 개시하게 되는 불안한 상황을 감소시킬 수 있다. 특히, 국방 분야에서는 특별한 상황이기에 소프트웨어의 매우 엄격한 테스트가 요구되므로[10], 보안에 취약한 프로그램 구조를 배제하려는 본 모델이 매우 유용할 것으로 생각된다.

그러나, 외주개발이 관리 상의 편의와 소프트웨어 관련 업무의 경감을 위해 추진되는 경우가 많은 현실에서, 발주처의 외주개발 주체의 소프트웨어 개발주기에의 관여는 관리의 복잡성과 외주개발 주체의 업무 증가를 불러올 수도 있기에 최대한 간결하면서도 효율적으로 추진되어야 함은 주지해야 할 것이다.

참고문헌

[1] 양환석, 유승재, “웹 어플리케이션 보안을 위한 가상화 기반 보안 모델”, 융합보안 논문지, 제14권, 제4호, pages 28 - 32, 한국융합보안학회, 2014. 6.

[2] Jack Danahy, "The ‘phasing-in’ of security governance in the SDLC", Network Security, Vol. 2008, Issue 12, pages 15 - 17, December, 2008.

[3] 홍진근, “스마트폰 환경의 응용 소프트웨어 개발 과정에서 보안정책 이슈”, 디지털융복합연구, 제10권, 제10호, pages 319 - 324, 한국디지털정책학회, 2012. 11.

[4] Kibria Khalil Rana and Syed Shams Uddin Ahmad, "Bringing maturity to test", Electronics Systems and Software. Vol. 3, Issue 2, pp. 32-35, Apr/May, 2005.

[5] 최준열, 김외철, 허성재, “국방 소프트웨어 개발 및 테스팅 현황”, 정보과학회지, 제32권, 제4호, pages 27 - 34, 한국정보과학회, 2014. 4.

[6] 김점구, 노시춘, “SSE-CMM 기반 기술적 보안 성숙도 수준 측정 모델 연구” 융합보안 논문지, 제12권, 제4호, pages 25 - 31, 한국융합보안학회, 2012. 9.

[7] 한경숙, 김태환, 한기영, 임재명, 표창우, “대한민국 전자정부 소프트웨어 개발보안 가이드 개선 방안 연구”, 정보보호학회논문지, 제22권, 제5호, pages 1180 - 1189, 한국정보보호학회, 2012. 10.

[8] 김동진, 정윤식, 윤광열, 유해영, 조성제, 김기연, 이진영, 김홍근, 이태승, 임재명, 원동호, “보안기

능의 무력화 공격을 예방하기 위한 위협분석 기반 소프트웨어 보안 테스팅", 정보보호학회논문지, 제22권, 제5호, pages 1191 - 1204, 한국정보보호학회, 2012. 10.

[9] 방지호, 하란, "소프트웨어 개발 보안성 강화를 위한 주요 보안약점 진단규칙 연구", 한국통신학회논문지, 제38권, 제10호, 한국통신학회, 2013. 10.

[10] 송경희, 이병걸, 류동국, 김진수, "국방 소프트웨어 시험성숙도모델 개발을 위한 테스트 프로세스 모델 비교분석", 한국멀티미디어학회지, 제11권, 제2호, pages 58 - 69, 한국멀티미디어학회, 2007. 6.

[저 자 소 개]

최 경 호 (Kyong-Ho Choi)

2002년 경기대학교 경제학사
2005년 경기대학교 경제학석사
2008년 경기대학교 정보보호학박사
2015년 (주)제이에프테크 연구소장

email : cyberckh@gmail.com

이 동 휘 (DongHwi Lee)

2007년 2월 경기대학교 정보보호(박사)
2011년~2012년 University of Colorado Denver, Dept. of Computer Science and Engineering
현재 동신대학교 정보보안학과 조교수

email : dhclub@dsu.ac.kr