기업용 하이브리드 애플리케이션 보안 위협 요소 및 대응방안 연구

이윤재*, 오준석**, 김용원***, 이봉규****
*연세대학교 전기전자공학부
**, ***연세대학교 방송통신정책연구소
****연세대학교 정보대학원
e-mail:{largowinch, jseok, kkjerry, bglee}@yonsei.ac.kr

Security Threat Factors and Improvement Methods in Enterprise Hybrid Applications

Yoonjae Lee*, Junseok Oh**, Yongwon Kim***, Bong Gyou Lee****
*School of Electrical & Electronic Engineering, Yonsei University
**, ***Communications Policy Research Center, Yonsei University
****Graduate School of Information, Yonsei University

요 약

본 연구의 목적은 차세대 MEAP 환경에서의 보안 평가 모델을 제시하는 것이다. 기업용 애플리케이션 개발환경은 PES 및 MEAP을 거쳐 완벽한 OSMP구현을 위한 HTML5 환경으로 발전하고 있다. 이와 더불어 보안의 위협도 증대되고 있으나, HTML5 환경에서의 보안에 대한 연구는 미흡한 실정이다. 이러한 문제에 대비하기 위해서는 기존 개발환경의 보안 특징을 살펴볼 필요가 있다. 본 연구에서는 보안위협요소를 Back-End System, Client, Developer, OS 4가지로 도출한 후, 이에 해당하는 보안 위협 문제들을 살펴보고 보안 평가 모델을 제시하였다. 본 모델은 단계별 보안이슈를 포함하고 있으며, 향후 HTML5 시대에 논의될 보안 이슈의 방향성을 제시한다는데 그 의미가 있다. 따라서 본 연구는 기업형 하이브리드 애플리케이션 개발을 준비하는 기업 및 연구자에게 시사점을 제공할 것으로 기대된다.

1. 서론

최근 아이폰의 등장과 함께 촉발된 스마트폰의 급속한 확산과 업무환경의 물리적 한계로부터 벗어나고자 하는 기업들의 욕구는 EMS(Enterprise Mobility Service) 구현의 필요성을 제기하였다. 스마트폰으로 이메일 확인이나 결재 정도만 가능할 것이라는 기존 예상과는 달리, 실제로 영업지원, 물류관리 등 현업 부서에서 사용되고 있으며, 경영진 역시 사내 업무에만 한정적으로 활용하는 것에서 벗어나 대외 비즈니스로도 확대 적용하기 시작하였다. 그러나 기업들은 새로운 플랫폼과 애플리케이션 버전이 등장할 때 마다 애플리케이션을 매번 새롭게 개발해야 하는 문제에 직면하였다. 자원과 인력의 낭비를 막기 위한 기업들의 통합 플랫폼에 대한 요구 또한 증가 하였다.

이러한 기업들의 요구에 따라 등장한 것이 바로 MEAP (Mobile Enterprise Application Platform)이다. 실 예로 가트너(Gartner)는 오는 2012년이면 모바일 오피스를 구현하려는 기업의 95%가 MEAP을 활용할 것이라고 전망하였다[1]. 이러한 환경 구축 시 가장 큰 장애요인이 바로 보안성이다. 삼성경제연구소의 조사에 따르면 '모바일 환경에서 가장 우려되는 요소가 무엇인가'에 대한 질문에 47.9%로 보안성을 꼽았다[2].

향후 HTML5의 표준화와 발맞춰 MEAP 또한 하이브리드 애플리케이션 개발환경으로 발전할 것으로 예상되고 있는 만큼 HTML5 웹 환경에서의 보안기술에 대한 연구도 필요한 상황이다.

이에 따라 본 논문에서는 차세대 기업용 애플리케이션 개발환경으로 각광받는 MEAP과 HTML5 기술에 대해 알아보고, 하이브리드 개발환경에서 고려해야 할 보안 기술의 취약점 및 대응방안에 대해여 연구한다.

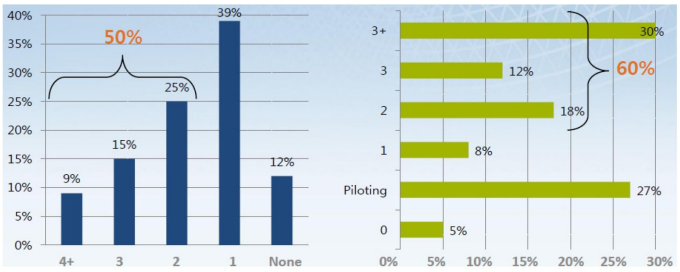
2. 관련연구

2.1 MEAP

모바일 앱 개발방식 중 하나로 대표되는 MEAP은 OSMP(One Source Multi Platform)의 핵심요소이다. OSMP란 애플리케이션을 하나의 소스(Source)를 통해 아이폰, 안드로이드 등과 같은 여러 플랫폼에서 사용가능하게 만드는 개발환경을 지칭한다.

다양한 스마트폰의 등장으로 인해 모바일 애플리케이션 개발, 유지보수 및 통합 측면에서 기업들은 심각한 문제를 안게 되었다. (그림 1)과 같이 모바일 업무환경의 다변화에 따라 기업들은 다음과 같은 특징을 갖는 솔루션을

"본 연구는 방송통신위원회의 방송통신정책연구센터운영지원사업의 연구결과로 수행되었음"(KCA-2011-0902-1)



(a) IT 부서에서 지원해야 할 모바일 OS의 수 (b) 모바일 서비스를 하고 있는 App.의 수

(그림 1) Mobile 업무환경의 다변화

*그림출처 : Forrester(좌), IDC(우)

필요로 하게 되었다[3].

- 다수의 스마트폰 OS, HW 및 통신사 환경에 대한 코드 재활용성 제공
- 플랫폼 공유를 통한 서비스 제공 인프라
- 미들웨어에 대한 중복 투자 방지
- 다양한 단말기에 대한 모바일 화면 개발 지원
- 테스트와 연동을 위한 지원 등의 편의성

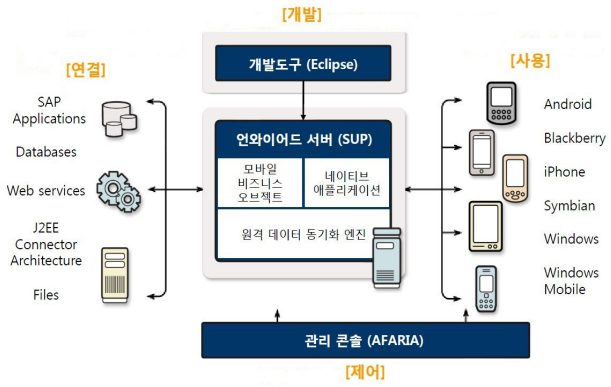
(그림 2)는 기본적인 MEAP 아키텍처의 예를 보여준다. MEAP 아키텍처는 크게 연결, 개발, 사용, 제어 파트로 구성되어 있다. 먼저 연결 부분은 이 기종의 백앤드 데이터 소스(ERP, CRM, Web 서버 등)를 MEAP 서버와 연결한다. 개발 파트는 Eclipse 등과 같은 애플리케이션 개발 툴을 통해 데이터 소스를 모바일 애플리케이션에 연동시킨다. 사용 파트는 MEAP 서버를 이 기종 단말기에 구현 가능하게 한다. 마지막으로 제어 파트는 관리 콘솔을 통해 기기 관리 및 보안 지원을 한다[2].

2.2 HTML5

모바일 웹 개발방식 중 하나인 HTML5는 웹 문서를 만들기 위한 프로그래밍 언어인 HTML(HyperText Markup Language)의 차세대 웹 표준안으로, 하나의 언어(Java Script), 하나의 데이터 모델(XML, DOM), 하나의 레이아웃(CSS)을 통일적으로 제공하여 텍스트, 오디오, 비디오, 그래픽 등을 통합 제공해준다[4].

HTML5가 주목받는 이유는 중요 서비스나 솔루션이 특정 브라우저에 종속됨으로써 발생할 수 있는 문제를 예방하고 모바일 브라우저에서도 PC와 동일한 서비스를 가능하도록 하는데 있다. 나아가 변화하는 인터넷 환경에 맞는 새로운 서비스를 제공하기 위한 기능들을 추가할 수 있으며, 어도비 플래시나 마이크로소프트의 실버라이트, SUN의 자바FX와 같은 플러그인 기반의 인터넷 애플리케이션에 대한 필요를 줄일 수 있다. 게다가 로컬 스토리지의 사용으로 오프라인에서도 애플리케이션의 역할을 수행할 수 있다.

HTML5은 모바일 애플리케이션 개발 분야에서 다음과 같은 파급 효과를 가진다.



(그림 2) MEAP 아키텍처

*그림출처 : SAP 자료 재구성

- 진보된 웹 표준 기술로 편리함을 제공하여 개발 비용의 절감
- 단말이나 플랫폼의 종류에 구애받지 않는 모바일 웹의 완성
- 모바일 웹을 기반으로 한 기기의 네이티브 기능 사용가능

2.3 모바일 앱(MEAP) 기반, 모바일 웹(HTML5) 기반

기업용 모바일 애플리케이션 개발은 크게 MEAP으로 대표되는 모바일 앱 개발방식과 HTML5로 대표되는 모바일 웹 개발방식으로 나뉘는 추세이다.

모바일 앱 방식은 단말기에 애플리케이션을 직접 설치하는 방식으로, 단말기의 특성을 반영하여 최적화된 UX(User Experience)를 제공 할 수 있다. 이 방식은 데이터를 단말기에 저장 하는 것이 가능하며, GPS나 음성인식 등 단말기의 네이티브 기능도 모두 활용 할 수 있어 성능, 보안 측면에서 모바일 웹 대비 강점을 가지고 있다. 하지만 이 방식은 플랫폼에 따라 개별로 개발되어야 하며, 같은 플랫폼이라도 버전이 나올 때 마다 개발 작업을 새로 해야 되는 문제점이 있다.

모바일 웹 방식은 웹 브라우저를 통해 애플리케이션을 구동하는 것으로 플랫폼에 제한이 없고, 제작과 유지보수의 편의성 때문에 모바일 앱 대비 강점을 가진다. 따라서 향후 HTML5가 표준화 될 2014년에는 모바일 웹 방식이 더욱 확고한 지위를 누릴 수 있을 것으로 보인다. 그러나 최적화 되지 않은 UX와 웹 방식이 가지는 고질적인 보안의 약점과 데이터베이스 동기화 문제는 해결되어야 할 과제이다.

위와 같은 개발방식에 의한 차이 때문에 다음과 같은 업무유형별 개발전략의 이원화가 필요하다.

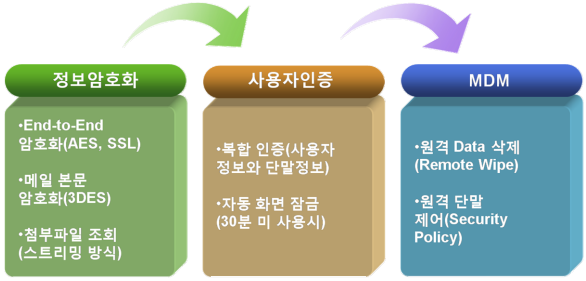
<표 1> 업무유형별 개발전략 이원화

분류	업무 유형	개발 전략
애플리케이션에 따른 개발전략 이원화	데이터저장, DB동기화	MEAP 기반 모바일 앱
	콘텐츠공유, 정보제공	HTML5 기반 모바일 웹
디바이스에 따른 개발전략 이원화	승인업무, 커뮤니케이션	스마트폰
	문서작성, 콘텐츠 생성	스마트패드

3. 보안 위협 요소 및 대응방안 분석

3.1 MEAP 보안 위협 요소

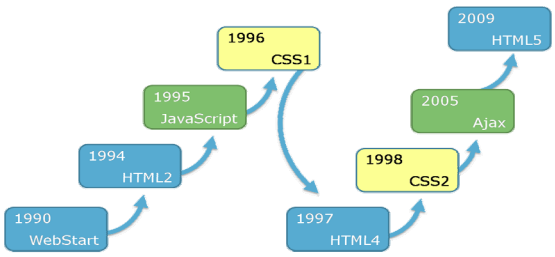
MEAP은 기본적으로 중앙 시스템 관리자가 보안시스템의 정책을 관리하는 중앙 집중식 보안시스템을 가진다. 이는 MEAP이 분실가능성이 높은 모바일 디바이스를 기반으로 하기 때문이다. 따라서 MEAP은 기존의 레거시가 가지는 보안 위협 요소와 더불어 모바일 디바이스를 사용함으로써 생기는 위협 요소까지 가지게 된다. 이러한 위협들에 대처하기 위한 MEAP의 보안요소는 (그림 3)에서 보는 바와 같이 정보암호화, 사용자인증, MDM(Mobile Device Management) 3가지로 분류할 수 있다.



(그림 3) MEAP 보안요소

이전의 기업운영 시스템(Previous Enterprise System)이 MEAP 환경으로 발전하면서 클라이언트로부터의 보안요소가 중요해졌다. 기업의 입장에서 클라이언트의 단말분실은 곧 기업정보유출로 이어질 수 있는 중대한 보안사고이므로, MEAP 보안은 MDM에 초점을 맞추고 있다.

3.2 HTML5 이전 보안 위협 요소



(그림 4) 웹 개발방식 발전과정

웹 개발방식은 (그림 4)와 같이 발전하여왔다. HTML5이 나오기 전까지는 Ajax 환경이 가장 많이 사용되었다. Ajax는 비동기식 통신을 사용하여 웹 사용자로 하여금 요청서비스에 대한 응답 속도 향상, 대기시간 불필요, 트래픽 감소, 비용절감의 이점을 가지고 있었다. 하지만 Ajax는 <표 2>에서 볼 수 있듯이, XSS, Injection 등의 보안 위협 요소에 취약한 모습을 보여주었다. 이러한 보안위협 요소들은 HTML5 환경에서도 웹 플랫폼의 고질적인 보안 취약점으로 작용하고 있다[5].

<표 2> HTML5 이전 보안 위협 요소

보안 위협 요소	설명
A1 - 크로스 사이트 스크립팅(XSS)	피해자의 브라우저 내에서 악성 스크립트를 실행하게 만듦
A2 - 인젝션 취약점	공격자가 삽입한 데이터에 대해 의도치 않은 명령어 실행
A3-악성파일실행	원격 파일 인젝션으로 심어진 악성 코드를 이용 서버 훼손
A4-불안전한 직접 객체 참조	내부 구현 객체에 대한 참조를 URL의 매개변수에 노출
A5-크로스 사이트 요청 변조(CSRF)	브라우저가 사전 승인된 요청을 취약한 웹 애플리케이션에 전송
A6-정보유출과 부적절한 오류처리	애플리케이션 자체의 문제점을 통해 개인정보 누출
A7 - 취약한 인증 및 세션 관리	자격 증명과 세션 토큰의 보안실패로 비밀번호, 키 손상
A8-불안정한 암호화 저장	웹 애플리케이션의 부실한 암호화 기능을 공격
A9-불안전한 통신	보안에 실패한 네트워크 트래픽을 가로챔
A10-URL접속통제 실패	허용되지 않은 URL에 직접 접속하여 악의적인 행동 수행

*표출처 : OWASP Top 10 2007 재구성

3.3 HTML5 이후 보안 위협 요소

HTML5는 기본적으로 웹 방식이 가지고 있는 보안 위협요소를 가지고 있다. 기존 Ajax 환경에서 HTML5 환경으로 변화하면서 생긴 가장 큰 변화는 실시간 웹을 구성하기 위한 Web Socket, 오프라인 애플리케이션 환경을 위한 로컬 스토리지를 들 수 있다. 이들은 HTML5를 통해 우리가 얻을 수 있는 큰 장점이다. 하지만 디바이스 상의 로컬 스토리지에 대한 보안 위협 요소 또한 새로운 이슈로 떠오르고 있다. HTML5 웹 환경에서 거론되는 보안 위협 요소는 대표적으로 다음의 5가지를 들 수 있다[6].

3.3.1 Cross-Document Messaging

HTML4의 경우 한 도메인의 페이지에서 다른 도메인의 접근이나, 데이터 교환 등을 허락하지 않는다. 이러한 HTML4의 보안 기능은 악의적인 사이트가 합법적인 사이트로부터 데이터를 가로채는 것을 막는다. HTML5의 경우 portMessage라는 새로운 API를 제공하는데, 이는 한 도메인에서 다른 도메인으로 데이터를 전달하는 스크립트에 대한 프레임워크이다. 이때 데이터 요청이 해킹인지 판단하기 위해 쓰이는 것이 Origin Check와 Object Property이다. 개발자는 Origin Check를 위해 Object Property를 포함시키는데, 만약 부주의한 개발자가 이를 간과한다면 해킹사이트에 portMessage 요청을 노출하게 된다.

3.3.2 Local Storage

로컬 스토리지 접근을 통해 웹 애플리케이션의 속도를 빠르게 만들 수 있다. 특히 동일한 데이터의 쿼리가 반복적으로 요구될 때 성능이 향상된다. 그러나 이것은 부주의한 개발자들에 의해 새로운 보안위협을 가져온다. 이메일

혹은 사용자 패스워드 등 개인적인 데이터를 오프라인 데이터베이스에 저장할 때, 부주의한 개발자가 SSL(Secure Socket Layer)과 준비된 SQL Statement를 사용하지 않는다면 해커들이 이를 가로채거나, SQL Injection 공격을 받게 될 가능성이 있다.

3.3.3 Attribute Abuse

HTML5에서는 새로 생긴 태그나 속성 등이 다양하게 존재한다. 이러한 속성 값들이 자동적인 스크립트 실행의 트리거가 될 경우 위협에 되게 된다. 한 예로, HTML5의 'Autofocus' 속성은 자동적으로 브라우저의 포커스를 특정한 요소로 이동시킨다. 해커들은 이 속성을 훔쳐서 악성 코드가 실행 되게 하는 윈도우로 포커스를 옮겨 버릴 수 있다. 이와 마찬가지로 'Poster'와 'Srcdoc' 속성도 악성 사이트에 의해 남용될 수 있다.

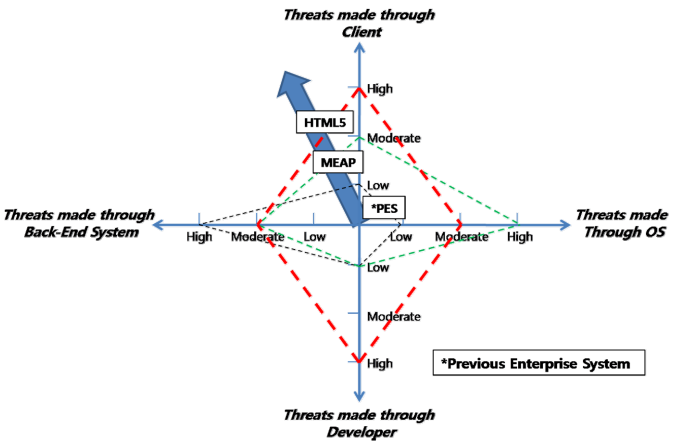
3.3.4 Inline Multimedia and SVG

HTML5는 더 이상 3rd Party 플러그인에 의존해 멀티미디어 재생을 할 필요가 없다. <audio>, <video>, <svg> 태그를 이용해서 미디어 포맷을 표현할 수 있다. 그러나 이런 점은 브라우저 개발자가 복잡한 멀티미디어 렌더링을 수행해야하는 부담으로 작용할 수 있으며, 보안 위협요소로 작용할 수 있다. 구글 크롬의 초기 버전에서 이미 SVG 피싱 버그가 보고 된 바 있다.

3.3.5 Input Validation

악의적인 사용자들은 열악한 사용자 Input을 이용해, 실행 코드 또는 다른 트리거를 훔쳐낼 수 있다. 서버의 관여 없이 페이지 자체에서 Input Validation을 빠르게 수행이 가능하다. 그러나 이런 기능수행이 개발자들에게 보안에 대한 잘못된 정보를 제공할 수 있다. 예를 들어, 잘못된 Regular Expression(regex)- 를 이용해서 DOS(Denial of Service) 공격을 할 수 있다.

3.4 보안 평가 모델 및 보안 위협 요소 대응방안



(그림 5) 제안된 보안 평가 모델

<표 3> HTML5 보안위협요소 대응방안

보안 위협 요소	대응방안
Cross-Document Messaging	Origin Check 포함
Local Storage	SSL, SQL Statement 사용 유일한 데이터베이스 이름 이용
Attribute Abuse	Autofocus, Poster, Srcdoc 속성 오남용 주의
Inline Multimedia and SVG	<audio>, <video>, <svg> 피싱 주의
Input Validation	Validation Code 생성 시 권한 주의

4. 결론 및 향후연구과제

MEAP은 완전한 OSMP를 위해 HTML5 환경으로 진화할 것이다. 하지만 HTML5 기술이 2014년에 표준화가 제정될 것에 비해 보안 대비책 확립은 미미한 실정이다. 본 논문은 아직 정립되지 않은 HTML5 보안 위협 요소들에 대해서 알아보고, 이 요소들이 MEAP 환경에서 미치는 영향을 분석하였다.

본 논문에서 제안 된 보안 평가 모델(그림 5)의 구조는 이봉규 외(2010)의 논문을 참고하여 구성하였다[7]. 보안 평가 모델의 각 축은 PES, MEAP, HTML5의 개발환경의 변화에서 공통적으로 발견될 수 있는 위협요소인 Back-End System, Client, Developer, OS로 나누었다. 각 환경에 대한 평가는 기존 논의를 바탕으로 하였다.

PES의 경우 Back-End System의 직접적인 해킹에 가장 큰 위협을 받는다. MEAP기반 환경의 경우 단말기기 분실(Client 측면), 모바일OS 자체의 허점(OS 측면)들이 강조된다. HTML5기반 환경의 경우 부주의한 개발자를 통한 CDM(3.3.1), Local-Storage(3.3.2)등의 위협(개발자 측면)과 악의적인 사용자를 통한 Input Validation 등의 위협(클라이언트 측면)이 강조된다.

향후에는 제안된 보안 평가 모델을 바탕으로 HTML5 환경으로 발전해 나가는 기업용 하이브리드 애플리케이션의 보안요소에 대한 기술적 연구가 뒷받침 되어야 할 것이다.

참고문헌

[1] Gartner, "Gartner Symposium 2010", 2010.

[2] 권용현, "성공적인 모바일 환경 구축을 위한 SAP 전략", SAP Korea, 2011.

[3] 김동한, "Enterprise Mobility 구현의 핵심, MEAP 동향", Nipa, 2011.

[4] 이은민, "HTML가 웹 환경에 미치는 영향", 정보과학회지, 제29권, 제6호, pp.55-60, 2010.

[5] 김미선 외 4명, "웹 2.0과 Ajax 보안 취약점", 정보과학회지, 제25권, 제10호, 2007.

[6] Aron Wesis, "Top 5 Security Threats in HTML5", eSecurity Planet, 2010.

[7] 이봉규 외 4명, "Developing Security Assessment Models in Web^2 Mobile Environments", Lecture Notes in Computer Science, Vol.6385, pp.73-84, 2010.