Internet and Information Security 제1권 제2호 (2010년 11월) pp.101~118

국내 인터넷뱅킹 계좌이체에 대한 MITB 취약점 분석

A Vulnerability Analysis of MITB in Online Banking Transactions in Korea

맹영재* 신동오** 김성호*** 양대헌**** 이문규*****
Young-Jae Maeng Dong-Oh Shin Sung-Ho Kim Dae-Hun Nyang Mun-Kyu Lee

초 록 인터넷뱅킹은 사용자가 자주 이용하는 중요한 금융 서비스 중 하나이다. 국내에서는 사용자의 컴퓨터에 설치되는 악성프로그램을 의식하여 ActiveX, BHO(Browser Helper Object) 형태의 보안 플러그인을 설치하고, 이들이 정상적으로 실행되고 있을 때만 인터넷뱅킹 서비스를 제공되도록 하고 있다. 그러나 현재 제공되고 있는 보안 프로그램은 거래내용이 담긴 HTML 문서 자원을 보호하지는 않고 있기 때문에 MITB(Man-in-the-Browser) 공격이 가능한 것으로 조사되었다. MITB 공격은 거래내용을 조작하고 이 사실을 사용자가 인지하지 못한 상태에서 거래를 승인하도록 유도하기 때문에 치명적인 속임수라 할 수 있다. 이 논문에서는 국내의 인터넷뱅킹 계좌이체에 대한 MITB 공격방법과 분석결과를 실제 실험을 통해 보이고, 문서변조 공격에 대한 두 가지 대응방법, CAPTCHA 응용방법과 추가장치 이용방법에 대해 분석한다.

주제어: 인터넷뱅킹, 계좌이체, 문서변조, 화면변조, MITB, 취약점분석

Abstract Internet banking is a well-known and crucial financial service for the management of user accounts. In Korea, the service is provided when security programs such as ActiveX and Browser Helper Object (BHO), are running correctly, to avoid a malicious program being installed in a user's computer. Nevertheless, we discovered that a Man-in-the-Browser (MITB) attack is practically feasible due to a shortage of current security programs to protect the document resources that contains transaction information. The MITB is a critical fraud because the user could be led to confirm a transaction that is manipulated by a malicious program, without noticing the fabrication attack. In this paper, we analyze the vulnerability through a real Internet banking example, and show two countermeasure approaches to this vulnerability, which are a CAPTCHA application and the use of additional devices.

Keywords: Internet banking, online transaction, document manipulation, screen manipulation, MITB, vulnerability analysis

논문접수일: 2010. 4. 22, 논문수정일: 2010. 11. 16, 게재확정일: 2010. 11. 19
이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임.(2010-0013254)

* 인하대학교 컴퓨터정보공학부 박사과정
(Doctoral Student, Dept. of Computer Science & Information Technology, Inha University, brendig@isrl.kr)
** 인하대학교 컴퓨터정보공학부 석사과정
(Graduate Student, Dept. of Computer Science & Information Technology, Inha University, mannershin@isrl.kr)
*** 인하대학교 정보통신대학원 석사과정
(Graduate Student, Dept. of Information Technology & Telecommunications, Inha University, night12th@isrl.kr)
**** 인하대학교 컴퓨터정보공학부 교수, 교신저자
(Professor, Dept. of Computer Science & Information Technology, Inha University, nyang@inha.ac.kr)
***** 인하대학교 컴퓨터정보공학부 교수
(Professor, Dept. of Computer Science & Information Technology, Inha University, mklee@inha.ac.kr)

Ⅰ. 서 론

국내에 인터넷뱅킹이 도입된 지 10년이 넘었다. 한국은행이 발표한 2010년도 2/4분기 인터넷뱅킹 서비스 이용현황에 따르면 인터넷뱅킹 이용자 수는 6,334만 명으로 1/4분기에 비해 2.8% 증가하였으며 자금이체서비스의 경우 일평균 406만 건, 약 29조 9,392억 원으로 전분기 대비 각각 3.6%와 3.0% 증가하였다. 금융서비스 전달채널 중에서 인터넷뱅킹을 통한 입출금 및 자금이체 거래는 34.1%, 조회서비스의 경우 66.1%를 차지하여 가장 높은 업무처리 비중(처리건수 기준)을 차지하고 있는 것으로 조사되었다. 인터넷뱅킹 중 모바일뱅킹 이용고객은 전분기말 대비 11.2% 증가한 1,318만 명, 자금이체서비스 이용 건수 및 금액은 각각 16.0%와 14.0% 증가한 44만 건, 4,078억 원으로 큰 폭의 증가율을 보이고 있다. 특히, 3G방식의 휴대전화에 설치하는 VM(Virtual Machine) 방식의 모바일뱅킹 등록고객이 크게 늘어나면서 가파른 증가율은 지속될 전망이다(한국은행, 2010).

인터넷뱅킹은 그 특성상 시간에 구애받지 않고 비대면 거래로 이루어는 서비스이기 때문에 컴퓨터가 사용자의 신원 또는 거래를 확인할 수 있도록 하는 인증 프로토콜의 보안성이 무엇보다 중요하다. 현재 인터넷뱅킹의 인증 프로토콜에는 사용자가 외우는 비밀, 사용자가 지니는 비밀 두 가지가 주로 사용되고 있으며 사용자가 지니는 비밀은 다시 고정된 비밀과 고정되지 않은 비밀로 구분된다. 사용자의 계정에 대한 패스워드, 공인인증서, 공인인증서 패스워드, 보안카드가 고정된 비밀에 속하며, 이들은 고정되어 있다는 특성 때문에 한 번 노출되면 사용자가 비밀을 바꾸기 전까지 공격자에 의해 재사용될 수 있다는 단점이 있다. 이와는 다르게 OTP(One Time Password) 발생기는 매번 다른 응답을 생성하고 각 응답은 유효시간이 제한되어 있을 뿐만 아니라, 한 번 사용되면 재사용이 불가능하도록 되어 있어 고정된 비밀에 비해 안전하다. 하지만 OTP가 적용된 시스템이라도 안심할 수는 없다.

공격자가 OTP와 같이 고정되지 않은 비밀을 악용할 수 있는 방법은 크게 두 가지로, 비밀을 가로채 공격자가 직접 사용하는 방법과, 비밀이 입력될 HTML 문서(이하 문서)를 변조하여 비밀을 공격자가 원하는 방향으로 악용하는 방법이 있다. 공격자가 피싱(phishing) 또는 파밍(pharming) 등 공격으로 사용자의 비밀을 얻어내는 방법이나 어깨너머 훔쳐보기(shoulder surfing, 사용자가 비밀을 조회한 순간부터 노출가능), 키로거(key logger, 시스템에 입력되는 순간부터 노출가능)를 통해 비밀을 얻어내는 방법 등이 전자에 속한다. 이러한 공격들이 MITM(Man-in-the-Middle) 형태로 응용되면 그 비밀의 유효시간 내에 또는 사용자보다 먼저 전송하여 공격자의 의도대로 사용될 수 있다.

사용자의 비밀을 악용하는 다른 방법으로는 비밀이 입력되는 문서를 변조하는 것이다(Gühring, 2007; Reed, 2008). 문서의 변조가 가능하다는 가정에서는 [그림 1]의 예제에서와 같이 사용자의 의심을 받지 않고 위조된 문서에 사용자의 비밀을 입력받을 수 있다. [그림 1]에서 사용자가 바라보는 내용은 문서원본에서 가져온 것이고 비밀이 입력되는 부분은 위조된 문서의 것이다. 사용자는 내용에 이상이 없음을 확인하고서 서명하겠지만 실제로 사인되는 문서는 사용자가 바라보는 내용과는 다른 공격자가 작성한 내용이 담긴 문서인 것이다. 문제는, 이렇게 사용자의 눈을 속이고 사용자의 의심없이 공격자가 원하는 내용으로 거래를 완료시키는 것이 인터넷뱅킹 계좌이체에서도 가능하다는 것과 그 공격이 자동화될 수 있다는

[그림 1] 공격자의 위조된 문서에 서명

것이다. 공격자가 인터넷뱅킹 계좌이체에 대해 문서변조 공격을 자동화하기 위해서는 계좌이체가 이루어지는 문서를 변조할 수 있어야 하고 공격자가 의도하는 거래정보를 문서에 강제로 입력할 수 있어야 한다. Ⅲ장에서는 이 두 과정에 대해 소개하도록 한다.

인터넷뱅킹 계좌이체에 대한 문서변조는 사용자가 '뱅킹 PC 지정서비스'[1]를 이용하더라도 공격자가 원하는 내용대로 거래를 성사시킬 수 있기 때문에 치명적인 공격이라고 할 수 있다. 이러한 공격을 피하기 위해서는 문서의 변조여부 즉, 문서 전체에 대한 무결성을 확인할 뿐만 아니라 실행중인 문서의 자원에 대한 외부 프로그램을 차단하고 악의적인 키 데이터 입력이 불가능하도록 해야 한다.

이 논문에서는 현재의 인터넷뱅킹 서비스에 대해 문서변조 공격이 가능하다는 것을 실험을 통해 보이고 그 취약점의 분석 및 대응방법에 대해 소개한다. 이 논문의 Ⅱ장에서는 관련연구 및 동향을, Ⅲ장에서는 국내 인터넷뱅킹 환경과 문서변조 및 수신계좌변조 가능여부를 살펴보고, Ⅳ장에서는 ㅁ은행에 대한 인터넷뱅킹 계좌이체에 대한 MITB 취약점을 분석하였다. Ⅴ장에서는 대응방법을 제시하고 Ⅵ장에는 결론을 담는다.

Ⅱ. 관련연구 및 동향

웹브라우저에 악성프로그램이 상주하여 공격하는 것은 Augusto(2005)에 의해 처음 발표되었고 Gühring(2007)에 의해 MITB라고 불리기 시작하였다. MITB은 사용자의 컴퓨터에 악성프로그램이 설치되어 있는 경우에 가능한 공격이다. 악성프로그램은 주로 웹을 통해 배포되는데, 이 악성프로그램 파일의 다운로드나 설치 전에는 사용자의 동의가 필요하다. 사용자 계정 컨트롤(UAC: User Account Control)이 적용된 윈도우 VISTA 이상에서는 프로그램을 설치하기 전에 사용자에게 한 번 더 프로그램의 설치 여부를 묻는다. 문제는 일반 사용자들의 지식이 전문적이지 않아서 프로그램의 설치확인 또는 동의과정에 나타나는 내용을 이해하기 힘들다는 것이다. Sharek et al.(2008)은 사용자가 진짜와 가짜 팝업 에러창을 어느 정도 구분할 수 있는지와 'OK' 버튼을 선택한 사용자들에게 그 이유를 물어보았다. 73%의 사용자가 가짜 팝업 에러창을 구별하지 못했고, 'OK' 버튼을 선택한 사용자 중에서 12%는 '에러 내용에서 그렇게 지시했기 때문에', 23%의 사용자는 '에러 메시지를 볼 때면 언제나 OK를 누른다', 42%의 사용자는 '에러창을 없애고 싶어서'라고 답한 것으로 조사되었다. 이 결과는 대부분의 사용자가 에러 메시지에 큰 관심을 기울이지 않고 있다는 사실을 보여준다. 국내 공공기관이나 금융서비스 등의 홈페이지에서 요구하는 ActiveX가 정상적으로 설치하지 않으면 그 서비스를 이용할 수 없도록 되어있다는 점을 고려하면, 사용자는 프로그램 설치 동의를 '예' 또는

1) 자신의 인터넷뱅킹 서비스를 지정된 IP 또는 컴퓨터에서만 가능하도록 하는 서비스

'아니오'의 선택문제이기보다는 서비스를 제공받기 위한 하나의 절차로 인식하고 있을 수 있다. 이러한 환경에서 악성프로그램이 어렵지 않게 배포될 수 있다는 사실을 예측할 수 있다.

Kim et al.(2010)은 한국의 인터넷뱅킹이 다양한 보안기술을 제공하고 있음에도 불구하고 더 나은 보안성을 기대하기 힘들다는 점을 SSL/TLS(Secure Sockets Layer/Transport Layer Security)와 OTP를 적용하고 있는 영국과 비교하여 보였다. 이 연구에서는 한국의 인터넷뱅킹이 피싱 공격에 대해 무방비하다는 것과 공인인증서의 문제점, 플러그인의 한계 그리고 공개되지 않은 인증 프로토콜에 대한 보안성에 대해 지적하였다. 피싱 공격과 관련해서는 외국 인터넷뱅킹의 EV-SSL(Extended Validation SSL)을 예로 들어 보였다. EV-SSL이 적용된 사이트에 접속하면 웹브라우저의 주소창 오른편에 웹사이트 ID가 표시된다. 이는 사용자가 접속하고자 하는 홈페이지에 정상적으로 접근했다는 것을 시각적으로 보여주는 것이기 때문에 피싱을 효과적으로 방지해준다.

비슷한 피싱 방지 방법으로는 야후의 '보안씰'이 있다. 보안씰은 사용자가 어떤 이미지나 문구

[그림 2] EV-SSL이 적용된 홈페이지의 예

를 등록하여 놓으면, 이후에 사용자가 야후의 로그인 화면에 접근했을 때 이 이미지를 보여준다. 피싱 공격은 사용자가 접근하고자 했던 시스템의 외부에서 이루어지기 때문에, 사용자가 접근한 홈페이지에 대해 사용자 스스로 올바른 홈페이지인지 여부를 판단할 수 있도록 시각정보를 이용하는 것은 훌륭한 접근방법이라고 할 수 있다.

하지만 보안씰은 이미지를 등록한 컴퓨터에서만 동작한다는 단점이 있고, EV-SSL 또한 사용

[그림 3] 야후의 보안씰

자가 이를 확실하게 인식하지 못할 경우에는 여전히 피싱 공격에 노출될 수 있다. 그 예로, 사용자가 '주소창이 초록색인 경우에는 안전하다'고 인식하고 있을 경우, EV-SSL이 아닌 단순 SSL을 적용한 피싱 사이트 또한 안전하다고 판단할 수 있다. 공격자는 StartSSL과 같은 곳에서 무료인증서를 비교적 쉽게 발급받아 악용할 수 있다.

또한 Kim et al.(2010)은 사용자가 정한 패스워드로 암호화되어 유지되는 공인인증서의 개인키 보안성에 대해 의문을 제기하였다. 추가로, 비밀키가 노출될 경우 공격자가 사용자의 권한을 남용하고 그의 의도대로 거래를 성사시킬 수 있다고 하였으나 이는 사실과 다르다. 공격자의 의도대로 거래를 성사시키기 위해서는 먼저 키보드 보안을 무력화하거나 우회할 수 있어야 하고, 사용자가 지니는 보안카드나 OTP와 같은 비밀정보가 있어야 한다. 결국 공인인증서의 비밀키가 노출되었을 경우에 공격자가 할 수 있는 일은 사용자의 거래

내역이나 정보를 제한적으로 조회하는 것으로 한정된다.

Ⅲ. 국내의 인터넷뱅킹 환경 및 문서변조 공격에 대한 취약점

ActiveX와 같은 플러그인으로 공인인증서를 사용하는 국내의 인터넷뱅킹 환경은 SSL/TLS를 주로 이용하는 외국의 인터넷뱅킹과는 다른 모습을 보인다. 이 장에서는 국내에서 계좌이체서비스를 보호하기 위해 사용되는 비밀(공인인증서와 보안수단), 비밀을 보호하기 위한 키보드 보안 프로그램, 문서의 암호화를 소개한다. 또한 문서변조 공격이 완료되기 위한 조건인 문서변조와 수신계좌 변경방법에 대해 알아본다.

1. 계좌이체에 사용되는 공인인증서와 보안수단

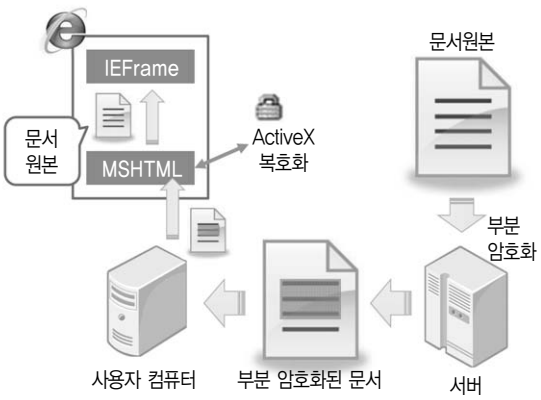
국내에서는 PKI(Public Key Infrastructure) 기반의 인터넷뱅킹 시스템을 이용하고 있다. 사용자가 인터넷뱅킹을 사용하기 위해서는 금융기관에 직접 방문하여 보안카드 또는 OTP 발생기를 수령하고, 이를 이용한 온라인 인증과정을 거쳐 공인인증서를 발급받아야 한다. 발급단계에서는 공인인증서 패스워드를 정하도록 되어 있는데, 이는 저장장치(하드디스크나 USB 등)에 보관될 공인인증서의 비밀키가 쉽게 노출되지 않도록 암호화하기 위함이고, 이 때문에 공인인증서를 사용하기 위해서는 매번 사용자에게 공인인증서 패스워드를 입력받아 복호화해야 한다. 이 패스워드는 사용자 스스로 정하는 것이기 때문에 Brute-force 또는 사전탐색 공격으로 노출될 수 있다. 공격자가 공인인증서와 공인인증서 패스워드를 소유하게 되면 인터넷뱅킹 시스템에 사용자의 권한으로 접근할 수 있으나, 이 권한으로 계좌이체 등의 거래는 발생시킬 수는 없고 사용자의 개인정보나 거래내역 등을 부분적으로만 조회할 수 있다.

계좌이체와 같은 거래를 하기 위해서는 보안카드나 OTP 같은 보안수단이 요구된다. 보안수단은 거래를 승인할 때 요구되며, 보안카드의 경우 질의응답(challenge-response) 형태로 사용하고 OTP의 경우 매번 다르게 생성한 응답을 사용하도록 하여 소유하는 비밀의 이용방법을 한층 강화한 것이다. 하지만 공인인증서와 보안수단을 사용한다고 해서 문서변조 공격에 안전한 것은 아니다. 공인인증서와 보안수단은 사용자가 외우고 있는 비밀(공인인증서 패스워드)과 소유하고 있는 비밀(공인인증서 파일, 보안수단)을 동시에 사용하는 2단계 인증인데, [그림 1]에서 보인 바와 같이 사용자를 안전하게 인증하였다는 것과 서버가 사용자의 비밀을 안전하게 전달받았다는 것이 사용자가 의도한 거래내용이 변조되지 않았다는 것까지 보장하지는 않기 때문이다.

2. 부분적인 암호화와 무결성 및 문서의 변조

국내의 인터넷뱅킹은 모든 패킷을 암호화하는 SSL/TLS와는 다르게 중요한 데이터만을 부분적으로 암호화하는 방식을 취하고 있다. [그림 4]와 같이 서버와 클라이언트에 설치되어 있는 플러그인이 세션키를 공유하고 서버가 중요한 데이터를 암호화하여 웹브라우저에 전송하면 자바스크립트가 암호화된 데이터를 플러그인을 이용하여 복호화한 뒤 출력하도록 하는 구조이다.

부분적으로 암호화된 부분은 세션키가 없으면 공격자의 의도대로 수정할 수 없지만, 문서 전체에 대한 무결성 확인 절차가 따로 존재하지 않는다면 암호화되지 않은 부분의 문서가 변조된 상태에서도 인터넷뱅킹 서비스가 정상적으로 이루어



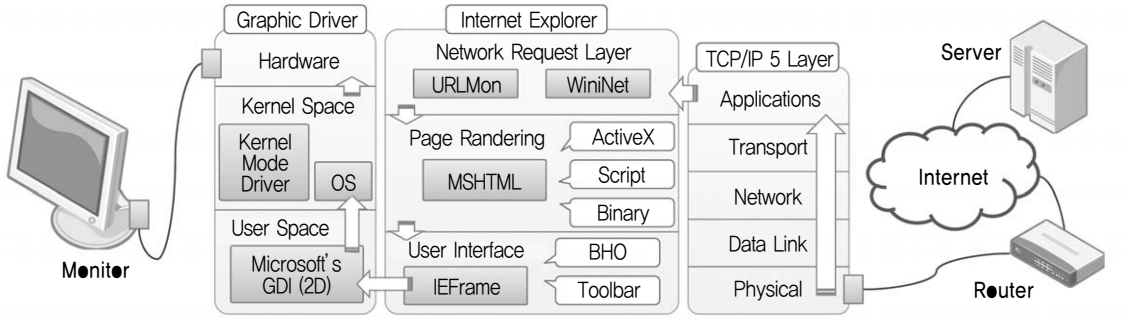
[그림 4] 국내 인터넷뱅킹의 부분 암호화 및 복호화 과정

질 수 있다. 이는 [그림 1]에서 보인 문서변조 공격의 가능성을 보여주는 것이다.

공격자가 문서를 변조할 수 있는 방법은 크게 세 가지로 나눌 수 있다. [그림 5]에서 네트워크 수준에서부터 문서가 IEFrame에 로드되기 전까지 구간에서의 DOM(Document Object Model) 개체 삽입을 통한 문서변조, IEFrame에 로드된 이후에 BHO(Browser Helper Object)나 Toolbar를 이용한 DOM 개체 삽입을 통한 문서변조, IEFrame에 로드된 이후에 또 다른 개체(IEFrame 또는 이미지)를 문서위에 덮어씌우는 방법이 그것이다. 첫 번째 방법의 경우, 문서가 IEFrame에 로드된 직후 보안 프로그램으로도 문서의 변조여부를 확인할 수 있지만, 두 번째 방법의 경우에는 거래가 발생하는 도중에도 문서가 변조될 수 있기 때문에 주기적으로 또는 특정 이벤트에 따라 문서의 변조여부를 확인해야 한다. 더욱 대응하기 어려운 공격은 세 번째 방법이다. 두 번째와 세 번째 방법의 다른 점은, 두 번째 방법의 경우 DOM 개체를 삽입하여 문서를 직접적으로 변조하지만, 세 번째 방법은 문서를 수정하지 않고서 다른 개체를 문서위에 덮어씌우는 방식이다. 즉, 문서의 변조여부만을 확인하는 방법은 문서를 직접적으로 수정하지 않는 세 번째 방법에는 대응할 수 없는 것이다. 아래에서 이 두 방법에 대해 더욱 자세히 살펴보도록 한다.

1) DOM 개체 삽입을 통한 문서의 직접적인 수정

인터넷뱅킹은 HTML과 각종 스크립트언어, ActiveX 그리고 BHO와 같은 추가적인 프로그램을 통해 서비스된다. 이중에서 특히, HTML 문서에 포함되어 있는 스크립트(Javascript 또는 VB script, 이하 스크립트)가 추가적인 프로그램의 설치나 통신을 제어하기 때문에 이 문서에 대한 무결성을 확인하는 것이 중요하다. 사용자의 컴퓨터에 악성프로그램이 설치될 수 있다는 가정에서



[그림 5] 서버에서 생성된 문서가 사용자의 화면으로 출력되기까지의 데이터 이동경로

또 하나 중요한 보안이슈는 실행중인 HTML 문서의 자원에 대해 악성프로그램의 접근을 차단하는 것이다. HTML 문서가 클라이언트에 수신될 때까지는 변조되지 않았다 하여도 수신이 완료된 이후에 악성프로그램 의해서 변조될 수 있기 때문이다. 문서변조가 가능하면 [그림 1]과 같은 공격을 구현할 수 있다. [그림 5]는 HTML 문서의 데이터 이동경로를 보여준다. 문서를 변조할 수 있는 구간은 네트워크 드라이버 수준에서부터 인터넷 익스플로러의 BHO에 이르기까지 다양하다.

스크립트가 DOM 구조의 HTML 문서에 접근하고 또 수정할 수 있다는 점은, 공격자가 HTML 문서의 한 부분에 스크립트를 삽입하여도 문서 전체를 변조시킬 수 있다는 것을 뜻한다. 예로, 악성스크립트를 BHO를 통해 문서의 하단에 삽입하거나, 악성스크립트를 문서의 상단에 위치시키고 HTML 문서 BODY의 onLoad 이벤트를 이용하여 그 스크립트를 호출할 수 있다. HTML 문서가 부분적으로 암호화되어 있어도 MSHTML. DLL(인터넷 익스플로러의 페이지 랜더링 부분)에 의해 해석될 때는 복호화된 후에 IEFrame상에 표현되기 때문에, 해당 부분을 자바스크립트가 DOM 구조로 접근하여 내용을 가져오거나 수정하는 것에는 아무런 지장이 없다. 이렇게 문서변조가 가능하다면 공격자는 사용자가 계좌이체 서비스를 이용할 때 [그림 1]에서와 같이 사용자의 비밀을 악용하여 계좌이체 가능금액 모두를 공격자의 계좌로 이체시키는 공격을 구상할 수 있다.

공격자의 악성스크립트는 사용자의 비밀(계좌비밀번호, 이체비밀번호, 보안카드, OTP, 공인인증서 패스워드 등)이 입력되는 폼은 수정하지 않는다. 수신자 정보(이체금액, 수신자 계좌번호, 수신자명)와 관련된 입력폼은 복사본을 생성한 이후에 이것이 화면에 원래의 이체정보 입력폼 대신 보이도록 상위계층으로 겹쳐놓는다.[2] 사용자는 복사된 가짜 입력폼에 이체정보를 입력하게 되고 이후의 모든 계좌이체 화면이나 이체내역과 관련된 화면에서는 사용자가 가짜폼에 입력한 정보를 출력하도록 하여 사용자가 문서변조 공격을 눈치 채지 못하도록 한다. 실제 계좌이체 정보가 반영되는 하위계층(화면에 보이지 않는 계층)에는 공격자의 계좌와 출금가능금액(출금가능금액 조회 메뉴를 통해 확인) 전부가 자동으로 입력되도록 한다. 비밀 입력폼은 원래의 것을 그대로 이용하기 때문에 인증 또는 승인에 필요한 비밀들은 사용자에 의해서 유효한 비밀이 정상적으로 입력된다.

2) 문서위에 개체를 덮어씌우는 간접적인 문서변조

문서를 직접적으로 수정하는 것이 불가능해질 경우, 공격자는 사용자를 속이기 위한 방법으로 문서위에 다른 개체를 올려놓는 방법을 생각할 수 있다. 그 개체는 공격자가 사전에 작성해놓은 이미지, 동영상 또는 IEFrame이 될 수 있으며, 이 개체는 계좌이체 과정에서 사용자를 속이고자 하는 시점에 출력되도록 작성된다. 실험에는 IHTMLDocument2로 문서에 접근하여 개체를 삽입할 시점을 탐지하고 Dialog 형태로 이미지를 덮어씌우도록 하였다. 이는 2절의 문서변조 방법에서 문서를 직접 수정하는 방법 대신, 화면에 실제로 출력되는 내용은 최상의 계층(layer)만 출력된다는 점을 악용하여 사용자가 바라보는 문서를 간접적으로 수정한 것이다.

위 두 가지 방법에 대한 실험결과는 〈표 3〉에 정리하여 놓았다. 더욱 자세한 계좌이체 과정과

2) 예로, CSS(Cascading Style Sheets)의 z-index 속성을 이용할 수 있다.

그 과정에 대한 문서변조 공격에 대한 취약점은 Ⅳ장에서 소개한다.

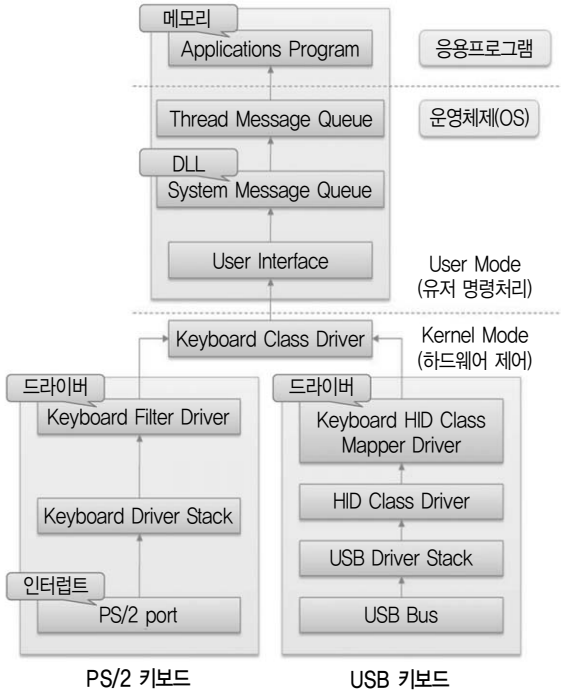
3. 키보드 보안 프로그램

한국의 인터넷뱅킹에서 설치하는 대표적인 보안 프로그램 중 하나로 키보드 보안 프로그램을 꼽을 수 있다. 키보드 보안 프로그램은 사용자의 컴퓨터에 키로거와 같은 악성프로그램이 설치되어 있을 수 있다는 가정 하에 사용자가 입력하는 비밀이 노출되는 것을 사전에 방지하기 위해 설치된다.

키보드는 PS/2 방식과 USB 방식이 있다. PS/2 방식의 경우 인텔 8042(i8042) 키보드 컨트롤러를 사용하며 키가 입력되면 일반적으로는 IRQ(Interrupt ReQuest)의 1번 채널을 이용하여 데이터를 전송하며 이후의 데이터 흐름은 [그림 6]과 같다. 키입력 정보를 가로채는 구간은 키보드 인터럽트 핸들러(keyboard interrupt handler)를 이용한 키보드 인터럽트 수준, 키보드 드라이버 수준, DLL Injection을 이용한 실행 코드의 메모리 영역 수준, BHO 등을 이용한 컴퓨터 메모리 영역 수준 등으로 다양하게 존재한다. 이러한 공격구간들에 대해서 국내의 키보드 보안 프로그램들은 각자의 구현 방법에 따라 [그림 6]에서 표현된 어떤 구간에서 키 입력 데이터를 AES 또는 SEED 암호화하여 보호하고 키로거가 입력값을 가로채지 못하도록 더미(dummy)값을 상위단계로 올려보내도록 되어있다. 또한, 키보드 보안 프로그램은 사용자가 입력하는 중요정보가 악성프로그램에 의해 노출되거나 변조되는 것을 방지한다.

1) 보안기능이 최소화된 키보드 보안 프로그램

하지만 키보드 보안 프로그램이 설치되지 않거나 정상적으로 동작하지 않는 사례가 있는데, 이



[그림 6] 키 입력 데이터의 전송 흐름 및 공격 구간

경우에는 인터넷뱅킹 서비스가 제공되지 않는다. 키보드 보안 프로그램 제작업체들은 이때에도 인터넷뱅킹이 가능하도록 사용자의 동의하에 보안기능이 최소화된(또는 Class ID만 같고 동작은 안 하는) 키보드 보안 프로그램을 제공한다(소프트포럼, 소프트캠프). 이 프로그램이 사용자의 컴퓨터에 설치되어 있으면 키보드 후킹(hooking)에 대해 안전하지 못하다. 문제는 공격자가 이 프로그램을 사용자 모르게 설치하여 키보드 보안 프로그램을 무력화하는 용도로 사용할 수 있다는 점이다. 키보드 보안 프로그램이 무력화되면 공격자는 사용자가 입력하는 비밀을 원격지에서 실시간으로 이용하거나 사용자 컴퓨터의 자원을 몰래 이용하여 공격자의 계좌로 이체를 시도할 수 있다. 이 논문에서는 기능이 최소화된 키보드 보안 프로그램에 대해서 더 이상 언급하지는 않지만, 이와 관련해서도 대비책이 마련되어야 할 것으로 보인다.

2) 악의적인 키 데이터 삽입

윈도우에서 제공하는 화상 키보드나 원격 데스크톱으로도 인터넷뱅킹이 가능하다는 사실에서 키보드 보안 프로그램이 적어도 인터럽트 처리 수준에서 동작하지 않는다는 것을 알 수 있다. 이러한 키보드 보안 프로그램은 인터럽트 후킹 수준의 키로거를 막을 수 없고, 국내 대부분의 인터넷뱅킹에 적용된 키보드 보안 프로그램의 경우에는 인터럽트 후킹 수준이 아니더라도 윈도우 API 수준의 keybd_event[3]로도 수신계좌번호를 입력하는 것이 가능했다. keybd_event는 키보드 드라이버의 인터럽트 핸들러가 호출하는 함수이다. 참고로, keybd_event는 윈도우XP 및 윈도우7 모두 정상적으로 동작하였지만 SendInput[4]은 윈도우XP에서만 동작하였다.

3) 계좌번호목록을 이용한 수신계좌번호 변경

키보드 보안 프로그램이 이상적으로 동작한다면 위에서 언급한 악의적인 키 데이터 삽입은 불가능해질 것이다. 하지만 그렇다 하더라도 문서변조 공격을 완전히 차단한 것은 아니다. 공격자는 스크립트 조작만으로도 사용자가 입력한 계좌번호를 취소하고 대신 계좌번호목록(자주쓰는 계좌번호 또는 최근입금 계좌번호, 이하 수신계좌목록)에 존재하는 어떤 계좌로 수신계좌를 변경하는 것이 가능하기 때문이다. 스크립트 조작을 통한 수신계좌 변경 실험결과는 다음에 정리하여 놓았다.

4. 문서변조 및 악성 키 데이터 입력 실험결과

본 절에서는 국내 주요 인터넷뱅킹에 대하여 문

〈표 1〉 보안 프로그램 종류 및 버전

약 어	보안 프로그램명	종류 및 버전
AOS	Ahnlab Online Security	프로그램: 2.1.3.2(Build 363) MyFirewall 방화벽 프로그램: 4.0.11.1(Build116) 방화벽 엔진: 2010.09.13.00 MykeyDefense 키보드 보안 프로그램: 2.5.31.1(Build 267)
XWC	XecureWeb ClientSM	XecureWeb ClientSM 4.1.2.4 Control Version 7.2.0.4
CKKP	Client Keeper KeyPro	3.0.2.1
INI	Inisafe Web v7.0 Client	InitechSHTTPInterface.10121.dll: 1.0.1.21 InitechSHTTPModule.10319.dll: 1.0.3.19 InitechSHTTPLocale.10035.dll: 1.0.0.35 InitechSHTTPTrayAgent.exe: 1.0.1.26 INIUpdater.ocx: 7.0.0.45 uninst.exe: 7.0.0.47
SCSK	SoftCamp Secure Keystroke	4.0
NPN	NProtect Netizen	nProtect Netizen: 2010.8.19.1 패턴파일: 2010-09-12.0

3) http://msdn.microsoft.com/en-us/library/ms646304(VS.85).aspx (2010. 4. 10 검색)
4) http://msdn.microsoft.com/en-us/library/ms646310(VS.85).aspx (2010. 4. 10 검색)

서변조와 수신계좌변조에 대해 2010년 9월 13일에서 16일 사이에 조사 및 실험한 결과를 제시한다. 보안 프로그램 종류 및 버전은 〈표 1〉에, 각 인터넷뱅킹에 적용된 보안 프로그램은 〈표 2〉에 정리하여 놓았다. 〈표 3〉은 각 인터넷뱅킹의 문서 암호화(SSL 또는 부분 암호화) 적용 여부, 문서 변조가능 여부 그리고 수신계좌 변조가능 여부를 조사 및 실험한 결과를 나타낸다.

실험에는 BHO를 이용하여 문서변조(DOM 개체 삽입, Dialog 개체 삽입), 수신계좌 변조(계좌번호목록 악용, 스크립트를 통한 키 입력, keybd_event를 통한 키 입력)를 시도하였다. 실험에는 인터넷 익스플로러8 32bit 버전이 사용되었다. 실험대상 모두 문서변조와 수신계좌 변조에 취약한 상태였으며, 이는 이들에 대해 공격자가 [그림 4]에서와 같이 자신의 의도대로 계좌이체를 완료시킬 수 있음을 의미한다.

Ⅳ. ㅁ은행에 대한 문서변조공격 실제사례

이 장에서는 사용자의 컴퓨터에 악성프로그램이 설치될 수 있다는 가정 아래, 문서변조 공격을 분석하여 보인다. 문서변조는 사용자의 컴퓨터에 설치되어 있는 악성프로그램이 계좌이체 페이지를 자동으로 변조하고, 이 변조된 문서에서 사용자가 어떤 수신자에게 계좌이체를 시도하면 이체가능금액 모두를 공격자의 계좌로 이체되도록 하는 공격을 뜻한다. 실험에서의 인터넷뱅킹에 대한

〈표 2〉 각 인터넷뱅킹에 적용된 보안 프로그램

은행명	키보드보안	기 타
A은행	SCSK	AOS, INI, VeraPort, u-safeon
B은행	SCSK	AOS, INI, NoPhishing
C은행	SCSK	NPN, XWC
D은행	SCSK	AOS, INI
E은행	CKKP	NPN, XWC
F은행	SCSK	AOS, INI
G은행	CKKP	AOS, XWC
H은행	CKKP	NPN, XWC

〈표 3〉 각 인터넷뱅킹의 문서변조 가능 여부 및 수신계좌변조 가능 여부 실험결과

조사 및 실험 결과 / 은행명	문서 암호화				문서 변조가능 여부				수신 계좌변조 가능여부
	SSL	부분암호	DOM 개체 삽입	Dialog 개체 삽입		계좌번호 목록악용	키 강제 입력		
							스크립트	keybd_event	
A은행	X	X	O	O	O	O	X	O	O
B은행	X	X	X	O	O	O	X	O	O
C은행	X	O	O	O	O	O	O	O	O
D은행	X	X	O	O	O	O	X	O	O
E은행	X	O	O	O	O	O	X	O	O
F은행	X	X	O	O	O	O	X	O	O
G은행	X	X	O	O	O	O	X	O	O
H은행	X	O	O	O	O	O	O	O	O

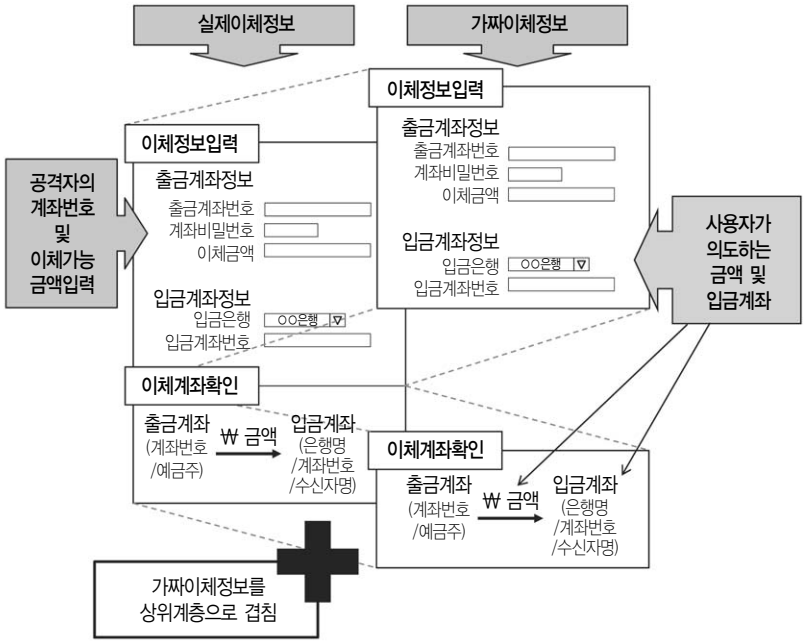
공격이 유효하다는 것은 〈표 3〉의 다른 은행에서도 동일하게 적용될 수 있다는 것을 의미한다.

1. 이체정보 입력화면

출금계좌를 선택하고 4자리 계좌비밀번호와 입금계좌번호 및 이체금액을 입력한다. 사용자가 키보드를 통해 입력하는 계좌비밀번호와 입금계좌번호에는 키보드 보안이 적용되어 있다.

1) 가짜 이체화면 출력

공격자의 관심은 입금계좌정보와 이체금액을 수정하여 이체가능 전액을 사용자 모르게 자신의 계좌로 입금시키는 것이다. 출금계좌정보(출금계좌번호, 계좌비밀번호)가 입력되는 곳은 수정하지 않고 사용자가 입력한 것을 그대로 이용하되, 입금계좌정보와 이체금액은 공격자의 의도대로 조작하도록 한다. 이를 위해서는 입금계좌정보와 이체금액 입력폼을 복사하여 CSS의 z-index 속성을 이용, 동일한 위치의 상위계층으로 설정한다. [그림 7]에서, 사용자가 보는 계좌이체 화면에서 입금은행, 입금계좌번호와 이체금액은 악성스크립트가 생성해낸 가짜 입력폼이고, 실제 계좌이체 정보는 하위계층의 입력폼에 위치한다. 이렇게 악성 스크립트가 변조한 계좌이체 화면과 원래의 계좌이체 화면은 표면상으로는 동일하기 때문에 육안으로는 구별할 수 없다. 가짜 입력폼을 작성한 후에는 이와 관련한 이벤트와 자바스크립트 함수들을 함수 오버라이딩을 통해 가짜 입력폼에서 동작하도록 수정한다.



[그림 7] 계좌이체 1, 2단계

2) 이체금액 조작

가짜 계좌이체 화면을 겹치는데 성공한 악성프로그램은 이체금액폼에 이체가능 전액을 입력한다. 이 과정은 이체가능 전액을 자동으로 입력해주는 '전액' 버튼을 악용할 수 있다. '전액' 버튼은 팝업창을 통해 사용자에게 전액이체 여부를 확인한 이후에 금액이 입력되도록 되어있다. 하지만 이 동작은 스크립트 수준에서 구현된 것이기 때문에 함수 오버라이딩을 통해 사용자 확인과정 없이 전액이 입력되도록 이용할 수 있다.

〈표 4〉 이체가능 전액을 자동으로 입력하는 자바스크립트

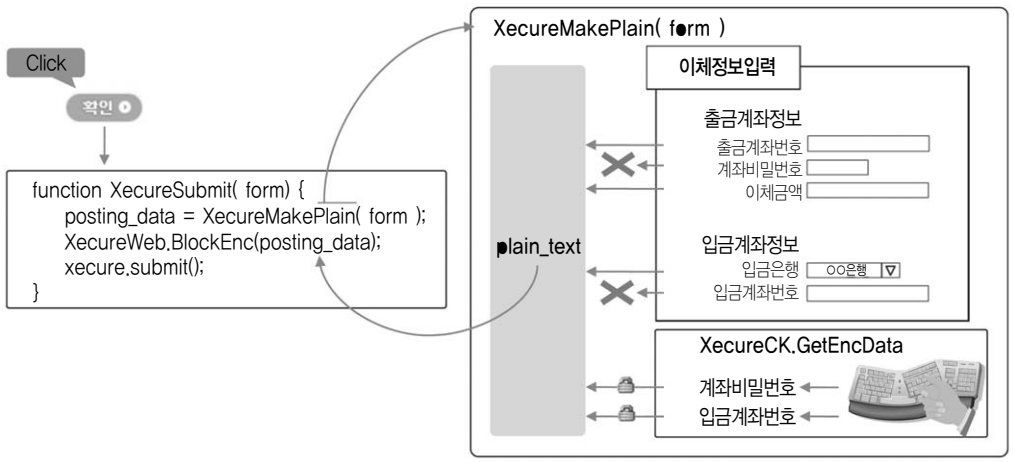
```
var index =
top.body.frm.WDRACTNO_15.selectedIndex;
with (top.body.document) {
  moneyFrmSum.ACTNO_15.value =
    frm.WDRACTNO_15[index].value;
  moneyFrmSum.action =
    "/pib/bank/fndt/wpdep011_28i.jsp";
  moneyFrmSum.target = "RECV_NAME";
  top.body.XecureSubmit(moneyFrmSum);
}
```

예로, 〈표 4〉는 이체가능 전액을 사용자 확인과정 없이 이체금액폼에 자동으로 입력하는 스크립트이다. 이 스크립트를 실행하면 이체가능 전액이 하위계층에 위치한 이체금액폼에 입력된다.

3) 입금계좌번호 조작

입금계좌정보는 입금은행과 입금계좌번호로 이루어져 있으며, DOM의 SELECT로 구성된 입금은행은 스크립트 수준에서 조작할 수 있다. 한편, 키보드 보안이 적용된(Enc 속성이 on인 폼) 계좌번호의 경우에는 키보드 보안 프로그램이 계좌번호를 직접 보관한다. 사용자가 계좌이체 확인을 클릭하면 스크립트가 키보드 보안 프로그램으로부터 암호화된 계좌번호를 [그림 8]과 같이 전송받고, 실제 계좌이체폼을 완성한 이후에 한 번 더 암호화하여 서버에 전송한다. 입금계좌번호에는 키보드 보안이 적용되어 있기 때문에 스크립트 수준에서 입금계좌번호를 조작하는 것은 쉽지 않아 보인다. 계좌비밀번호에도 키보드 보안이 적용되어 있지만 이 값은 그대로 이용할 것이기 때문에 조작할 필요가 없다.

하지만 자주 쓰는 입금계좌나 최근 입금계좌 목록에서 선택하여 입금계좌번호를 선택하는 방법은 자바스크립트 이벤트로 동작한다. 이 목록의



[그림 8] 계좌이체 확인과정에 사용되는 함수들과 그 동작과정

어떤 계좌를 이용하는 것은 스크립트 수준으로도 충분하다는 뜻이다. 〈표 5〉는 자주 쓰는 입금계좌의 계좌정보를 자동으로 입력하는 스크립트를 보여준다. E2E_FLAG.value가 '1'이면 키보드로 입력된 입력계좌번호를 뜻하고 '2'이면 자주 쓰는 입금계좌번호를 뜻한다.

〈표 5〉 자주 쓰는 입금계좌번호를 자동으로 입력하는 스크립트

```
with (top.body.document) {
 obj = getElementById( 'RCVACTNO_15' );
 obj.value = '공격자가 의도한 계좌번호' ;
 top.body.CKKeyPro_Clear("frm",
"RCVACTNO_15");
 frm.E2E_RCVACTNO_15.value = '공격자가 의도
한 계좌번호' ;
 frm.E2E_FLAG.value = "2";
}
```

그러한 리스트에 공격자의 계좌가 이미 존재하고 있었다는 것은 사용자와 어떤 관계가 있음을 의미한다. 즉, 이 리스트를 조작한 공격은 그 공격대상이 제한적일 수밖에 없다는 것이다. 공격대상범위에 제한을 받지 않기 위해서는 키보드 보안이 적용된 경우에도 입금계좌번호를 조작할 수 있어야 한다. 이 과정에는 앞서 언급하였던 keybd_event와 같은 키 입력 함수를 악용할 수 있다. 실험 결과, 이 함수는 〈표 3〉의 모든 은행에서 사용 가능하였다. 키보드 보안 프로그램이 이러한 함수를 불가능하도록 한다면 보안성은 향상될 수 있지만 화상키보드나 원격데스크톱으로 키 입력이 불가능해지기 때문에 편의성에 악영향을 주게 된다. 태블릿PC와 같이 터치 이벤트만이 존재하는 장치에서는 인터넷뱅킹이 불가능해진다는 것이다.

4) 사용자가 입력한 이체정보 유지

조작된 계좌이체가 공인인증서를 통해 승인받기까지 의심받지 않으려면, 사용자에게는 사용자가 입력한 이체정보를 계속 보여주어야 한다. 따라서 가짜 입력폼([그림 7]의 상위계층)에 사용자가 입력한 계좌정보를 악성프로그램에 유지하도록 한다.

2. 수신자명을 통한 입금계좌 확인

앞의 이체정보 입력화면에서 확인버튼을 누르면 입력한 계좌정보가 서버에 전송되고 입금계좌정보가 확인되면 이체정보(수신자명 포함)가 화면에 다시 출력된다. 이체추가를 통해 여러 건의 계좌이체를 추가할 수 있다.

1) 수신자명 조작

계좌이체를 서버에 요청할 때는 수신자명을 입력하지 않는다. 사용자가 가짜폼에 입력한 입금계좌에 대한 수신자명을 알 수 있어야 이 단계에서 사용자를 속일 수 있는 것이다. 이 단계에서 사용자를 속이는 것은 스크립트 수준에서 구현될 수 있다.

스크립트는 사용자가 입력한 입금계좌번호를 실제로 서버에 질의하여 수신자명을 알아내도록 한다. 수신자명을 얻어내기 위해 서버에 질의한 순간, 해당 입금계좌는 예비계좌이체로 지정되기 때문에 악성프로그램이 그냥 무시할 수는 없다. 때문에 수신자명을 얻은 이후에는 해당 계좌이체를 취소하도록 서버에 해당 계좌이체를 취소 요청한다. 이 행위가 누적되면 언젠가는 서버가 공격을 알아챌 수 있으므로 계좌이체를 취소하지 않고 실제로 전송하는 방법 또한 병행하도록 한다. 이 계좌이체는 사용자가 입력한 금액 그대로 이체되도록 하거나 스크립트가 임의대로 적은 금액을 이체하도록 할 수도 있다. 이체계좌확인 화면에는 사용자가 입력한 입금계좌에 대한 정보만이 출력되도록 한다.

3. 공인인증서를 통한 이체승인

입금계좌 정보(은행, 계좌번호, 수신자명, 금액)의 확인과 그 아래 보안카드번호 또는 OTP를 입력하도록 되어 있다. 보안카드 또는 OTP를 입력한 이후에는 공인인증서의 사용을 통해 최종적으로 계좌이체에 대한 사용자의 승인을 받는다. 공인인증서를 사용하는 화면의 상단에는 계좌이체 정보가 표시되는데, 계좌이체가 두 건 이상인 경우에는 계좌이체 정보가 나타나지 않는다.

1) 이체내용 조작

공격자는 거래내역이 변조된 상태에서 승인이 되도록 하기 위해 두 방법을 선택할 수 있다. 첫째, 공인인증서 창에 나타나는 이체내역을 수정한다. A은행의 경우에는 submitCert 함수를 통해 공인인증서 창에 출력할 이체내역이 출력되기 때문에 이 함수를 오버라이딩하여 공격자의 계좌정보가 아닌 사용자가 입력한 계좌정보를 출력하도록 수정할 수 있다. 둘째, 두 건 이상의 계좌이체를 수행하도록 하여 이체내역을 띄우지 않도록 한다. 두 건 이상의 계좌이체가 발생할 때는 이체내역이 나타나지 않는다. 이렇게 이체내역이 변조될 수 있는 상황에서는 공인인증서가 사용되었다 하더라도 사용자 입장에서는 부인방지 보안 서비스가 정상적으로 보장되지 않는 것이다.[5] 이 공격이 서버의 취약점으로 인해 발생한 것은 아니기 때문에 은행입장에서는 부인방지를 주장할 수 있다.

4. 이체결과

1) 이체결과의 조작

악성프로그램은 사용자가 입력했던 이체정보에 대한 이체결과를 출력하도록 한다. 특히, 계좌이체 후 잔액은 사용자가 의도했던 금액만큼만 제외하고 출력하도록 한다. 이후의 조회화면 또한 조작하면 사용자가 다른 채널을 통해 계좌를 조회하지 않는 이상 공격사실을 인지하지 못하도록 할 수 있다.

거래이용수단에 따른 보안등급별 거래이용수단 및 이체한도는 〈표 6〉과 같다. 이 논문에서 보인 MITB 공격은 사용자가 사용하는 거래이용수단이 OTP 발생기를 사용하는 1등급이라 하여도 공격자의 의도대로 계좌이체가 이루어지기 때문에 치명적이라 할 수 있다.

〈표 6〉 거래이용수단에 따른 보안등급

거래이용수단	보안등급
OTP 발생기 + 공인인증서	1등급
HSM[1] 방식 공인인증서 + 보안카드	
보안카드 + 공인인증서 + 2 channel 인증	
보안카드 + 공인인증서 + 휴대폰 SMS(거래내역통보)	2등급
보안카드 + 공인인증서	3등급

*출처: 금융위원회(2007)

주: 1) HSM: 공인인증서 복사방지를 위해 사용하는 보안성이 강화된 스마트카드, USB 저장장치

5) 사용자가 승인을 한 것은 사실이지만 사용자가 변조된 이체내역에 대해 승인하고자 했던 것은 아니다.

〈표 7〉 전자자금이체한도(지급이체의 경우, 단위: 억 원)

구 분				보안등급		
				1등급	2등급	3등급
현금카드	인출한도	1회	0.01			
		1일	0.06			
	이체한도	1회	0.06			
		1일	0.3			
텔레뱅킹	개인	1회		0.5	0.2	0.1
		1일		2.5	1	0.5
	법인	1회		1	0.2	0.1
		1일		5	1	0.5
인터넷뱅킹	개인	1회		1	0.5	0.1
		1일		5	2.5	0.5
	법인	1회		10		
		1일		50		
모바일뱅킹		1회	1			
		1일	5			
메일뱅킹		1회	0.1			
		1일	0.5			

*출처: 금융위원회(2007)

V. 대응방법

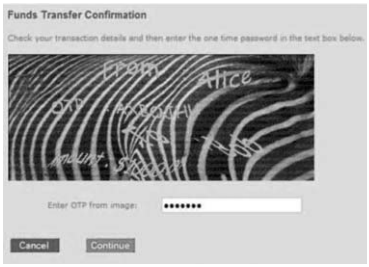
문서변조 또는 화면을 변조하는 공격에 대한 대응방법은 크게 두 가지로, CAPTCHA에 거래내력을 포함시키는 방법과 신뢰할 수 있는 다른 채널을 이용하여 거래내역을 확인하도록 하는 방법이 있다.

1. CAPTCHA를 이용한 부정승인 방지방법

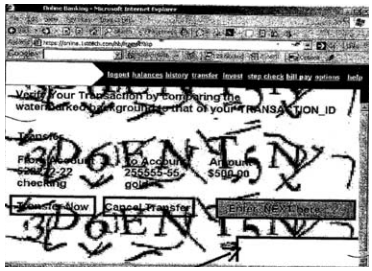
거래내역을 포함시킨 형태의 확장된 CAPTCHA는 자동화된 악성코드에 대응할 수 있지만 구현방법에 따라 악성코드에 취약할 수 있다. ArcotVPS (Gopalakrishna, 2009)의 경우 배경과 문자열의 색이 다르고 이미지에서 문자가 차지하는 비율이 낮다는 사실을 근거로 Kmeans++를 사용하면 문자열을 분리시킬 수 있다. 악성프로그램이 캡차를 읽을 수 없더라도, 추출된 4개의 문자열 중 하나를 정해 추출한 다음 변조된 거래내역에 대한 승인이미지를 생성할 수 있는 것이다. 따라서 악성프로그램은 CAPTCHA를 읽을 수 없어도 1/4의 확률로 공격에 성공할 수 있다.

MS 워터마크(Steeves & Snyder, 2007)는 일반 폰트로 거래내역이 표현된 배경 위로 CAPTCHA가 표현되어 있다. 이 기법에서는 CAPTCHA의 폰트가 더욱 두껍고 크기 때문에 윤곽선 검출(edge detection)의 Canny 필터(가우시안 마스크를 통해 잡영 제거 후 Sobel 필터 적용)를 통해 강한 윤곽선만을 남긴 후에 이를 추출해내면 CAPTCHA 문자열을 분리시킬 수 있다. 특정 두께에 대한 영역을 세선화(thinning) 하여 문자열을 얻어낸 다음, 이를 다시 Blur 필터 등으로 두껍게 만들면 더욱 선명한 문자열을 얻어낼 수 있다. 분리된 캡차 이미지는 변조된 거래내역에서 악용될 수 있다. 위의 두 기법은 CAPTCHA를 인식하고 그에 대한 응답을 사용자가 수동적으로 입력해야 하도록 되어 있다.

문맥 기반 CAPTCHA에서는 사용자에게 전용되는 CAPTCHA 이미지 중에서 사용자의 거래와 관련된 것만을 선택하도록 하였다(김성호 외, 2009). 이 기법에서 6개의 콘텐츠(수신은행, 수신계좌, 수신자명, 이체금액, 송신자, 송신은행)와 44개의 더미 캡차로 구성되어 있을 경우, 무차별 선택에 대한 보안성은 $1/(50\times\cdots\times45)=8.74\times10^{-11}$이다. 공격이 가능한(사용자가 입력한 수신은행과 이체금액은 그대로 사용) 최소한의 콘텐츠 변경은 수신계좌와 수신자명 두 가지를 변경하는 것으로, 이때의 보안성은 $1/(50\times49)=4.08\times10^{-4}$이다. 이 기법에서는 사용자가 여러 개의 CAPTCHA를 인식해야 하기 때문에 편의성이 높지는 않다. 하지



[그림 9] ArcotVPS



[그림 10] MS 워터마크

[그림 11] 문맥기반의 CAPTCHA

만 사용자가 CAPTCHA를 읽고 입력하는 형태가 아닌, 선택하는 용도로 사용하였기 때문에 더욱 복잡도가 높은 CAPTCHA를 이용할 수 있다. 예로, 심하게 뒤틀려 일부 영역이 인식하기 힘든 형태가 되어도 CAPTCHA를 선택하는 데에는 지장이 없다.

2. 부가장치를 이용한 승인방법

부가장치를 이용한 승인방법은 부가장치에서 거래내역을 출력하고 이 장치를 통해 승인하도록 하는 것이 목적이다. 현재 사용 중인 부가장치를 이용한 승인방법으로는 인터넷뱅킹 계좌이체의 전화승인서비스가 유일하다. 전화승인서비스는 계좌이체 승인단계에서 ARS 시스템을 이용해 사용자에게 전화를 걸어 거래내역을 확인시켜주고 거래를 승인하거나 취소할 수 있도록 하였다.

트랜잭션 서명기법은 사용자 토큰이라 불리는 추가적으로 발급받은 기기에 거래내역의 일부를 입력하면 토큰에 출력된 MAC을 사용자가 컴퓨터에 수작업으로 입력하는 방법이다(Hiltgen et al., 2008). 이 연구에 기초한 임형진 외(2008)는 사용자 토큰에 인증서를 저장한 형태 또는 사용자가 입력한 MAC을 컴퓨터에서 공인인증서를 통해 사인하는 방법으로 부인방지를 추가적으로 제공한다. 트랜잭션 서명기법은 사용자 입장에서는 일부이기는 하지만 거래내역을 한 번 더 입력해야 하기 때문에 편의성이 떨어진다.

IBM의 ZTIC은 USB 장치의 디스플레이를 통해 거래내역을 출력하고 두 개의 버튼을 통해 승인 또는 취소를 결정하도록 한다(IBM, 2008). ZTIC은 컴퓨터에 연결되면 USB 저장장치로 인식됨과 동시에 사전에 정의되어 있는 은행사이트로 연결되도록 하는 프록시를 설정하고, 웹브라우저와 해당 사이트와의 통신은 모두 ZTIC을 통해 이루어지며, 이 세션을 암호화하는 TLS/SSL의 키는 악성프로그램이 접근할 수 없도록 되어 있다. 내용을 확인하고 버튼으로 승인여부를 결정하면 되기 때문에 편의성이 좋지만 추가적인 장치를 발급해야 한다는 단점이 있다.

부가장치를 통해 승인하는 방법은 보안성은 높지만 거래 당시에 추가적인 장치를 소지하고 있어야만 승인이 가능하기 때문에 사용자 편의성이 떨어진다고 할 수 있다.

Ⅵ. 결 론

이 논문에서는 국내 인터넷뱅킹에 대한 MITB 취약점을 분석하여 보았다. 사용자의 컴퓨터에 공격자가 작성한 악성 BHO가 설치되어 있다면, 계좌이체를 할 때 이체가능 전액이 공격자가 의도한 계좌로 사용자 모르게 이체될 수 있다는 사실이

확인되었다. 이 공격은 사용자가 고정 PC에서만 계좌이체가 되도록 신청하고 은행에서 요구하는 모든 보안 프로그램이 정상적으로 동작하는 상태에서도 유효하다는 점, 그리고 사용자가 사용하는 거래이용수단의 보안등급이 높을수록 금전적인 피해가 더욱 커질 수 있다는 점에서 치명적이다.

키보드 보안 프로그램이 이상적으로 동작한다면 공격자가 의도한 계좌로 강제로 이체하는 것은 크게 제한된다. 공격자의 계좌가 사용자의 계좌목록에 유지되고 있어야 하기 때문이다. 입금계좌번호 입력이 키보드 보안 프로그램에 의해 제한되더라도, 공격자는 복호화된 사용자의 공인인증서를 통해 시스템에 접근하여 자신의 계좌를 사용자 계정의 계좌목록에 추가한 뒤에 공격할 수 있다. 공격자는 자신의 계좌가 사전의 등록되어 있는 사용자를 목표삼아 공격할 수도 있기 때문에 문서변조 공격은 여전히 적지 않은 보안문제가 될 것이다. 사용자의 편의성이나 프로그램의 원활한 동작을 위해 화상키보드, 원격데스크톱을 이용한 인터넷뱅킹이 허용된다면 이 논문에서 보인 문서변조 공격을 막는 것은 쉽지 않을 것으로 사료된다. 그러한 프로그램이 사용하는 이벤트 또는 함수를 공격자 또한 이용할 수 있기 때문이다.

서론에서 소개하였듯이, 최근에 모바일뱅킹의 이용자 및 이용도가 급증하고 있다. 사용자의 컴퓨터에 악성프로그램이 설치될 수 있는 인터넷뱅킹과 마찬가지로, 모바일뱅킹에서도 악성 애플리케이션이 설치될 수 있다는 가정 하에 보안 프로그램 등을 설치한다면 이 논문에서 보인 공격자의 가정이 모바일뱅킹에서도 성립하게 된다. 인터넷뱅킹에서의 MITB와 같은 공격이 모바일뱅킹에서 동작하는 형태로 진화할 수 있으므로 이와 관련된 대책이 시급하다고 할 수 있다.

▶ 참고문헌

금융위원회 (2007. 1). 『전자금융감독규정 시행』. 금융위원회 보도자료.

김성호 · 강전일 · 김기태 · 양대헌 (2009). 문맥기반의 캡차를 이용한 신뢰성 있는 인터넷 계좌이체 방법. 『한국정보보호학회 동계학술대회 논문집』, 19(2), 319-323.

소프트캠프 (2010a). 소프트캠프 키보드보안 바이패스 페이지. 2010. 8. 10 검색, http://www.softcamp.co.kr/scsk/bypass/

소프트포럼 (2010b). KeyPro 장애지원. 2010. 8. 10 검색, http://keypro.clientkeeper.co.kr:8080/faq/safe.jsp

임형진 · 이정근 · 김문성 (2008). 안전한 인터넷뱅킹을 위한 트랜잭션 서명기법에 관한 연구. 『인터넷정보학회 논문지』, 9(6), 73-79.

한국은행 (2010. 7). 『2010년 2/4분기 국내 인터넷뱅킹서비스 이용현황』. 한국은행 보도자료.

Augusto, P. B. (2005, September 15). O futuro dos backdoors - O pior dosmundos. Retrieved August 10, 2010, from http://www.paesdebarros.com.br/backdoors.pdf

Gopalakrishna, R. A. (2009). Authentication using a turing test to block automated attacks. U.S. Patent, 0 199 272 A1.

Gühring, P. (2007, January 24). Concepts against man-in-the-browser attacks. Retrieved August 10, 2010, from http://www2.futureware.at/svn/sourcerer/CAcert/SecureClient.pdf

Hiltgen, A., Kramp, T., & Weigold, T. (2006). Secure Internet banking authentication. *IEEE Security and Privacy*, 4(2), 21-29.

IBM (2008). ZTIC: Zone trusted information channel. Retrieved August 10, 2010, from http://www.zurich.

ibm.com/ztic/

Kim, H., Huh, J. H., & Anderson, R. (2010, January). On the security of Internet banking in South Korea. Retrieved August 10, 2010, from http://www.comlab.ox.ac.uk/files/2916/RR-10-01.pdf

Reed, B. (2008, January 14). New trojan intercepts online banking information. Retrieved August 10, 2010, from http://www.networkworld.com/news/2008/011408-silentbanker-trojan.html

Sharek, D., Swofford, C., & Wogalter, M. (2008). Failure to recognize fake Internet popup warning messages. *Human Factors and Ergonomics Society*, 52(6), 557-560.

Steeves, D. J., & Snyder, M. W. (2007). Secure online transactions using a CAPTCHA image as a watermark. U.S. Patent 0 005 500 A1.