Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.25, NO.4, Aug. 2015 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2015.25.4.941

자바스크립트 변조를 이용한 국내 인터넷 뱅킹 키보드 암호화 모듈 우회 공격

이 성 훈,[1,2†] 김 승 현,[2‡] 정 의 엽,[1,2] 최 대 선,[2] 진 승 헌[2]
[1]과학기술연합대학원대학교, [2]한국전자통신연구원

An Attack of Defeating Keyboard Encryption Module using Javascript Manipulation in Korean Internet Banking

Sung-hoon Lee,[1,2†] Seung-hyun Kim,[2‡] Eui-yeob Jeong,[1,2]
Dae-seon Choi,[2] Seung-hun Jin[2]
[1]University of Science and Technology,
[2]Electronics and Telecommunications Research Institute

요 약

인터넷의 발달과 함께 인터넷 뱅킹은 널리 사용되고 있다. 이와 함께 금전적 이득을 목적으로 인터넷 뱅킹을 타겟으로 하는 피싱 공격이 기승을 부리고 있으며, 이에 대응하기 위해 은행에서는 보안 모듈을 사용자 컴퓨터에 설치하고 인터넷 뱅킹을 이용하도록 요구한다. 하지만 메모리 해킹 공격 등 고도화된 피싱 공격은 보안 모듈을 우회한다. 본 논문에서는 국내 은행에서 제공하는 인터넷뱅킹 보안 모듈 현황을 알아보고, 그 중에서도 이체 정보의 암호화를 담당하는 키보드 보안 모듈을 분석한다. 그리고 자바스크립트를 이용하여 이체정보를 변조하는 기법을 제안한다. 키보드 보안 모듈은 공격자가 심어놓은 악의적 프로그램으로부터 사용자가 입력한 이체 정보의 노출 및 변조를 방지하는 기능을 담당한다. 하지만 우리가 제안한 자바스크립트 변조 공격은 키보드 암호화 모듈을 우회하고 이체 정보를 변조하는 것이 가능하다. 또한 이체정보확인 페이지를 변조하여 사용자가 정상적인 거래를 한 것처럼 속일 수 있다.

ABSTRACT

Internet banking is widely used in our life with the development of the internet. At the same time, phishing attacks to internet banking have been increased by using malicious object to make unfair profit. People using internet banking service in Korea is required to install security modules such as anti-virus and keyboard protection. However phishing attack technique has been progressed and the advanced technique such as memory hacking defeats the security module of internet banking service. In this paper, we describe internet banking security modules provided by Korean internet banks and analyze how keyboard encryption module works. And we propose an attack to manipulate account transfer information using javascript. Although keyboard protection module provides two functions that protect leakage and manipulation of account transfer information submitted by users against the malicious program of hackers. Our proposed technique can manipulate the account transfer information and result html pages.

Keywords: Internet banking, Keyboard protection, Key encryption module, javascript manipulation, Phishing attack

접수일(2015년 6월 18일), 수정일(2015년 8월 3일),
게재확정일(2015년 8월 4일)

† 주저자, sunghoon1130@etri.re.kr
‡ 교신저자, ayo@etri.re.kr(Corresponding author)

I. 서 론

국내 인터넷 뱅킹은 2000년 이후 인터넷의 발달로 인해 많은 사용자들이 사용해오고 있다. 한국은행의 자료에 따르면 2014년말 인터넷 뱅킹 등록고객수는 1억 319만명으로 전년말대비 8.1% 증가하였고, 2014년중 인터넷뱅킹 이용건수 및 하루 평균 금액은 6,645만건, 36조 8,550억원으로 전체 거래량의 22%를 차지하고 있다[1]. 이렇게 인터넷 뱅킹이 활성화 되고 많은 사용자가 이용함에 따라 사용자의 ID와 패스워드나 거래 정보를 탈취하여 악의적인 거래를 시도하는 피싱(phishing) 공격이 발생되어 왔고, 피싱 기법이 점점 고도화되고 있다.

국내 17개 시중은행의 최근 3년간 피싱 관련 금융사고 현황 자료 Table 1. 에 따르면 2012년 1만 7813건, 867억, 2013년 1만 9357건, 937억원, 2014년 2만8097건 1589억원으로 매년 피해건수와 피해금액이 증가하고 있다[2]. 기존의 인증정보나 거래 정보를 탈취하는 피싱 공격에서 액티브 피싱 공격, 파밍을 거쳐서 최근에는 메모리 해킹 공격이 문제되고 있다[3,4].

본 논문에서는 사용자가 정상 은행 사이트에 접속하여 인터넷 뱅킹을 시도하더라도 사용자가 입력한 이체정보를 자바스크립트 수준에서 변조하여 메모리 해킹과 마찬가지로 보안 모듈을 우회하고, 사용자에게 피싱 피해를 야기할 수 있는 이체정보 변조 방법을 제안한다.

논문의 구성은 다음과 같다. 2장에서는 국내 인터넷 뱅킹 이체 프로토콜을 분석한다. 3장에서는 인터넷 뱅킹 이체 정보 변조와 이에 대한 대응 방안 관련 논문들을 살펴본다. 4장에서는 본 논문에서 제안하는 방법에 대해서 토론하고 이에 대한 대응 방안과 한계점을 알아본다. 마지막으로 5장을 통해 이 논문의 결론과 향후 연구를 싣는다.

II. 인터넷 뱅킹 이체 프로토콜 분석

이번 절에서는 인터넷 뱅킹 이체 흐름을 분석하고 어느 부분에서 변조가 발생 가능한지를 설명한다. Fig. 1. 은 인터넷 뱅킹 이체 흐름을 보여준다. 사용자가 인터넷 뱅킹을 위해 로그인을 한 이후에 이체를 시도하는 부분부터 이체 흐름이 시작된다.

1. 사용자는 인터넷 뱅킹 사이트 메뉴에서 이체를 클릭하여 이체를 시도한다.
2. 은행에서 이체입력 html페이지를 사용자 컴퓨터에 전달하고 컴퓨터는 이체 입력화면을 보여준다.
3. 사용자는 컴퓨터에 입금계좌 비밀번호, 입금액, 수신계좌번호를 입력하고 확인 버튼을 클릭한다.
4. 컴퓨터는 이체정보가 제대로 입력됐는지 확인하고, 입금계좌비밀번호와 수신계좌번호의 암호값을

Table 1. Status of phishing attack of Korean banks[2]

Financial	2012		2013		2014		Total
Company	Cases	Money	Cases	Money	Cases	Money	Cases(Money)
Nonghyup	2,709	13,902	4,969	24,841	6,833	40,105	14,511(78,840)
Kookmin	5,700	26,713	3,755	18,257	4,105	22,802	13,560(67,772)
Shinhan	2,654	12,403	3,755	15,792	5,038	29,885	11,447(58,080)
Woori	2,600	11,528	2,582	10,068	4,344	19,438	9,526(41,034)
Hana	1,087	5,452	1,446	8,507	1,806	10,043	4,339(24,002)
Kiup	891	5,331	1,063	6,272	1,710	9,425	3,664(21,028)
KEB	430	2,088	487	2,206	1,228	6,992	2,145(11,286)
SC	406	1,891	378	2,176	1,003	6,129	1,787(10,196)
Citi	212	1,118	139	707	261	1,360	612(9,423)
Daegu	418	2,201	269	1,699	578	3,872	1,265(7,772)
Busan	387	2,342	186	1,257	349	2,572	922(6,171)
Kyongnam	120	575	71	317	220	1,907	411(2,799)
KwangJu	76	475	93	463	202	1,269	371(2,207)
Chonbuk	52	275	44	255	147	966	243(1,496)
Jeju	29	304	2	28	30	158	61(490)
Suhyup	30	140	62	623	90	959	182(1,722)
KDB	12	51	56	299	153	1,057	221(1,407)
Total	17,813	86,789	19,357	93,768	28,097	158,938	65,267(339,495)

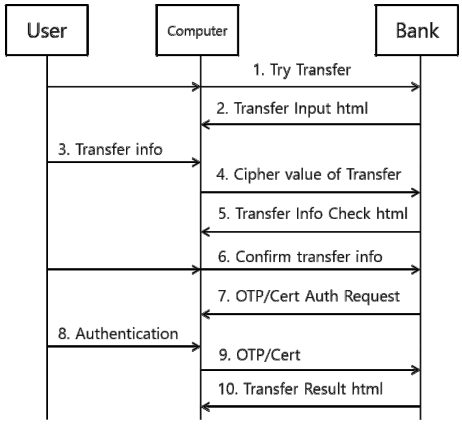

Fig. 1. Flow of account transfer of internet banking

은행에 전송한다.

5. 은행은 암호값을 확인하고 수신계좌번호의 수신자명을 포함한 이체정보확인 html 페이지를 전달하고 컴퓨터는 이체정보확인 화면을 보여준다.
6. 사용자는 이체 정보가 맞는지 확인한 뒤, 확인 버튼을 클릭하여 이체 흐름을 계속 진행한다.
7. 은행은 사용자 인증을 위해서 OTP, 공인인증서 서명을 요구한다.
8. 사용자는 OTP와 공인인증서 비밀번호를 컴퓨터 웹 브라우저에 입력하고, 확인 버튼을 클릭한다.
9. 사용자 컴퓨터의 웹 브라우저는 OTP와 공인인증서 서명값을 은행에 전송한다.
10. 은행은 OTP와 공인인증서 서명값을 확인 후, 이체를 완료하고, 이체처리결과 html 페이지를 사용자 컴퓨터에 전송한다.

위의 과정을 거쳐서 이체가 완료된다. 사용자가 입력한 이체 정보를 암호화하여 전달하는 과정인 Fig. 1. 의 4번에서 이체정보 변조 공격이 가능하다. Fig. 2. 는 각 html 페이지에서 변조되는 부분을 보여준다. 암호값 변조 이후에도, html 변조를

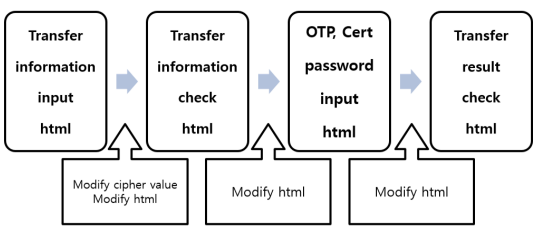

Fig. 2. Flow of internet banking manipulation

통해서 사용자가 원래 입력한 수신계좌번호를 보여준다. 이는 사용자에게 수신자명과 수신계좌번호가 변조되었음을 숨기기 위함이다. 이체정보의 암호값 변조와 html 변조에 대한 상세한 설명은 다음 절에서 한다.

2.1 이체정보 암호화

이체정보 암호화는 키보드 보안 모듈이 제공하는 기능이다. 키보드 보안 모듈의 주요 기능은 사용자가 입력한 값의 암호화 및 키로거 탐지이다. Table. 2. 는 국내 주요 은행의 키보드 보안 모듈을 정리한 내용이고, 크게 3개 회사 제품을 사용하고 있다. 본 논문에서는 3개 회사 제품 중 한 개 회사 제품을 선택하여 분석하였다.

사용자가 입력한 출금계좌번호 비밀번호와 입금계좌번호 등 이체 정보를 암호화 하는 과정은 Fig. 3. 의 4번 함수를 통해 시작된다. DataSubmit(frm) 함수를 호출하면, 키보드 암호화 모듈 라이브러리를 이용하는 자바스크립트 문서인 EncryptionKey.js (가칭)가 암호화를 진행한다. frm은 이체 페이지의 전체 프레임을 저장하고 있다.

EncryptionKey.js에서 GetEncDataFun() 함수가 실제 암호화 라이브러리를 호출하여 사용자가 입력한 출금계좌번호의 비밀번호와 입금계좌번호를 암호화한다.

Table 2. Keyboard security product of Korean major banks[5-7]

Bank Name	Product	Develop Company
KB Kookmin	Secure KeyStoke 4.0	SoftCamp
Woori	TouchEnKey	Raon Secure
Nonghyup	Secure KeyStroke 4.0	SoftCamp
Standard Charted	AOS(AhnLab Online Security) Anti-Keylogger	AhnLab
Hana (Open Banking)	TouchEnKey with E2E for 32bit	Raon Secure
Hana	Secure KeyStroke	SoftCamp
Shinhan	Secure Keystroke 4.0	SoftCamp

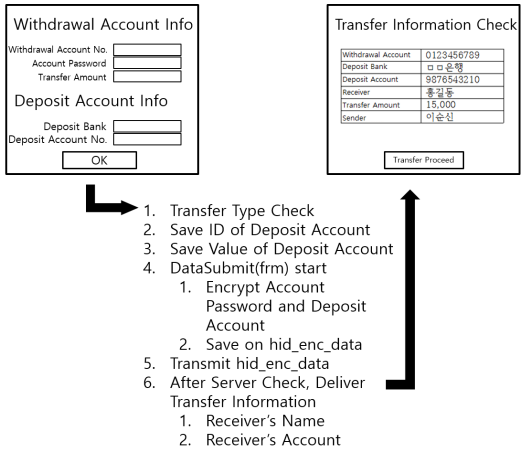

Fig. 3. Detailed flow of account transfer

GetEncDataFun은 3개의 인자("", frm.name, frm.elements[36].name)를 입력값으로 받는데, 마지막 인자인 frm.elements[36].name이 실제 암호화될 데이터이다. frm.elements[36]은 사용자가 입력한 입금계좌번호 입력창의 ID값이다. 입력창의 ID값은 매번 이체거래 진행시에 서버로부터 값을 받아오기 때문에 항상 다른 값이다. 그래서 전체 프레임에서 엘리먼트 값을 인자로 사용한다. 산출된 암호값은 hid_enc_data에 저장된다.

hid_enc_data에는 Fig. 4. 와 같이 출금계좌비밀번호(E2E_ACTPWNO_8) 암호값과 입금계좌번호(E2E_n1IL64P7Z5)의 암호값이 각각 입력창의 ID값과 함께 암호화가 됐다는 의미의 "E2E"를 붙여서 저장되고, %TK%로 각각을 구분한다.

이체번호 암호값은 이체번호 전체에 대한 암호값이 아닌 이체번호 하나하나에 대한 32비트 암호값으로 계산된다. 즉, 이체번호에 0123456789를 입력하면 320비트의 암호값이 계산된다. Fig. 5. 와 같이 32비트씩 분류하면, 각 숫자별 암호값을 획득할 수

```
<input type="hidden" name="hid_enc_data"
id="hid_enc_data" value=
"E2E_ACTPWNO_8=ce348c11b8af20cd36a7cf09d63b88bbd7
63b95a5dabf258eba3744212637a1f418ca7e54e45863031826
5bf96abd1ab298665f914b7236d236e006ec5a1f1b0%TK%
E2E_n1IL64P7Z5=ce348c11b8af20cd36a7cf09d63b88bbd76
3b95a5dabf258eba3744212637a1f7e8328cddfc2d9f42c63ee
3f146ea5bced5c339f6500228ad11a2709cb6335df0b8e9a950
d0ea7924e0a35fcf5ba532aea6cd2ee39974d805bc5fe5df42b
88a97121dfcc5494ef80959c6b99ff5a5a4664eba2cc06d5dae7
59d64a4ff5e3c2b0b3a99b658496907f93066592cf731469f8d1
be062033126609ef13f2c1c749ec88e0fd9c6811cf94edc24222
9dfabb1333586873e608370125f6c00a21da4cdd9e1452e52f2
97120da291ae5824c0b34%TK%">
```

Fig. 4. Example of hid_enc_data

```
E2E_q1DEbY7Hg1=
1e2dc965aa8805f0d9aeba28c42bffed → 0
e3fb98a481d2fd694952e35c0cecaaef → 1
d88d608dde13deee23137043b2d2c353 → 2
4d3846ea04a87556d0a38de61424e585 → 3
c61308a0c3d2d535d81ac7ca53b7adf5 → 4
f437b31f86ae4597155d7e3e1a26b74a → 5
71a4145faf94875c9fb9643baedffecf → 6
d41d4e09b64c8423a8cb437924477cca → 7
08ad5857e7ac1028d1f52f07f5b04fc4 → 8
1ba1c02ad254229ca2ac7af25e56e560 → 9
```

Fig. 5. Example of encryption value of account

있다. 이렇게 획득한 암호값을 재조합하면 공격자가 원하는 계좌번호의 암호값을 생성할 수 있다.

2.2 입금계좌번호 암호값 획득

앞 절에서 사용자가 입력한 번호에 대한 hid_enc_data에 저장된 암호값을 재조합할 수 있음을 확인했다. 이를 바탕으로 EncryptionKey.js에 삽입될 입금계좌번호의 암호값을 변조하는 자바스크립트는 Fig. 6. 과 같이 구성하였으며, 흐름은 아래와 같다.

1. 사용자가 입력한 이체번호에 대한 암호값이 저장된 hid_enc_data의 값에서 이체번호에 대한 암호값만을 추출한다. 예를 들어서 Fig. 4.에서 이체번호 첫 번째 숫자의 암호값은 32자리로 구분하면 "ce~bb" 까지이다.
2. 추출된 값을 32비트씩 사용자가 입력한 이체번호의 숫자수 만큼 반복하며 잘라내어 임시로 배열에 저장한다.
3. 사용자가 입력한 이체번호 암호값이 저장된 배열에서 공격자가 원하는 숫자의 암호값을 불러와서 재조합한다. 재조합된 암호값은 Fig. 6.의

```
var totalEnc = hid_enc_data.value.substring(161);
var n = new Array(10);
var cutlength = 32;
var trnsNum = frm.elements[36].value;
var num = new Array(trnsNum.length);
for ( var a = 0; a < trnsNum.length; a++){
        for (var b = 0; b < 10; b++){
                if (trnsNum[a] == b){
                        n[b] = totalEnc.cut(cutlength);
                        totalEnc = totalEnc.substring(cutlength);}}}
var changedEnc;
changedEnc =
n[1]+n[2]+n[3]+n[4]+n[5]+n[6]+n[7]+n[8]+n[9]+n[0]+n[1]+n[2]+
n[3]+"%TK%";
var manipulatedEnc = hid_enc_data.value.cut(161);
manipulatedEnc += changedEnc;
hid_enc_data.value = manipulatedEnc;
```

Fig. 6. Javascript code for manipulation of encryption value

changedEnc에 저장된다.

4. 재조합된 암호값을 hid_enc_data의 value에 저장한다.

Fig. 6. 의 자바스크립트는 GetEncDataFun() 함수의 마지막 부분에 추가된다. 사용자가 입력한 이체번호의 암호값이 계산된 후, 재조합을 해야하기 때문이다. 공격자가 원하는 입금계좌번호의 암호값으로 재조합된 이후에는 Fig. 3. 의 5번과 동일하게 이체정보의 암호값이 서버로 전달된다.

2.3 이체정보 변조 실행 화면

입금계좌번호의 암호값이 변조되면, 사용자가 입력한 웹 브라우저의 이체번호입력창의 번호가 바뀌는 것이 아니라, 메모리에 저장되어 있는 암호값이 변조된다. 하지만, 서버에서는 사용자가 웹 브라우저에 입력한 입금계좌번호와 서버에 전송된 암호화 모듈이 산출한 입금계좌번호의 암호값을 비교하여 검증하지 않는다. Fig. 7. 의 상단 그림과 같이 사용자가 입력한 이체번호로 진행을 하여도 암호값을 변조하여 변조된 이체 번호를 서버에 전송한다. 그리고 서버로부터 Fig. 7. 의 하단 그림과 같이 변조된 이체번호에 대한 이체정보확인 html을 받는다.

사용자가 입력한 입금계좌번호 값으로 정상 이체진행중이라고 속이기 위해, 은행 서버로부터 받은 이체정보확인 html의 내용을 변조해야 한다. html 결과 화면 변조는 DOM을 이용하여 html 내용을 변경한다[8]. 먼저, 사용자가 입력한 입금계좌번호 값을 사용자가 최근에 전송한 계좌이체번호 목록에서 검색하여 수신자명을 알아낸다. 만약 계좌이체번호 목록에 사용자가 입력한 입금계좌번호에 해당하는 수신자명이 없는 경우, 은행 서버로 입력정보를 전송하여 수신자명을 확인할 수 있다[9]. 수신자명과 입금계좌번호를 innerHTML을 이용하여 결과 화면을 변조한다.

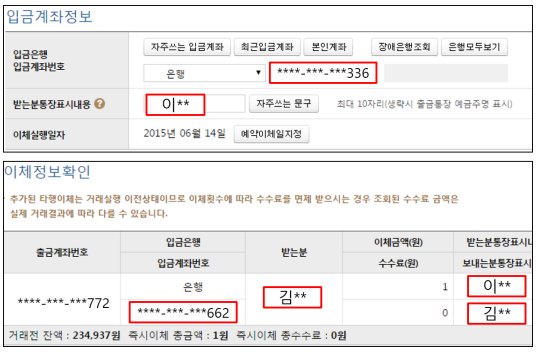

Fig. 7. Screen of account number typed by a user and manipulated account number

이체정보확인 html에서 수신자명과 수신계좌번호는 Table 3. 에서와 같이 tables71, 76, 78에 정보가 있고, OTP와 공인인증서인증을 요구하는 html 화면과 이체결과확인 html의 테이블 정보는 Table 4. 와 같다. Fig. 8. 의 자바스크립트를 이용하여 html 내용 변조가 가능하다. 흐름은 아래와 같다.

1. 태그이름으로 검색하는 document.getElementsByTagName("td")을 이용하여 html 구성 테이블 정보를 저장한다.
2. 저장된 테이블 정보에서 변조하고자 하는 테이블 번호를 찾아서 수정한다.
3. 수정된 내용을 해당 html 구성 테이블에 덮어쓰기로 수정된 내용이 표시되도록 한다.

Fig. 8. 의 자바스크립로 변조된 화면은 Fig. 9. 와 같다. 사용자에게 사용자가 입력한 이체정보로 진

```
javascript:tables=document.getEle
mentsByTagName("td");
tables[76].innerHTML="<td>1234
-123-123456</td>";
tables[71].innerHTML="<td><stro
ng><em>홍길동
</em></strong></td>";
Tables[78].innerHTML= " <td>홍
길동</td>";
document.getElementsByTagNam
e("td").innerHTML=tables;
```

Fig. 8. Javascript for manipulation of html value

Table 3. Table information in transfer confirmation html

Table No.	Value
69	Withdrawal Account No.
70	Deposit Bank
76	**Deposit Account No.**
71	**Receiver**
72	Transfer Amount(Won)
77	Fee(Won)
73	Recipient's Bankbook Indication Details
78	**Displayed Information on My Account**

행되고 있다고 속이기 위해서 Fig. 8. 의 자바스크립트를 이용한다. Fig. 9. 의 상단은 공격자에 의해 변조된 이체정보의 이체정보확인 html 페이지이고, 하단은 Fig. 8. 을 실행시킨 후의 html 변조 화면이다. 강조된 부분인 입금계좌번호와 받는분과 보내는분 통장표시내용 3군데가 변조됐다.

2.2절의 입금계좌번호 암호값 획득 및 암호값 변조 공격과 html 내용 변조 공격을 이용하여 사용자가 입력한 계좌번호가 아닌 공격자가 변조한 계좌로 실제 입금이 된다. 수신자는 본인이 입력한 수신계좌번호로 정상적인 거래가 완료됐다는 메시지를 보게

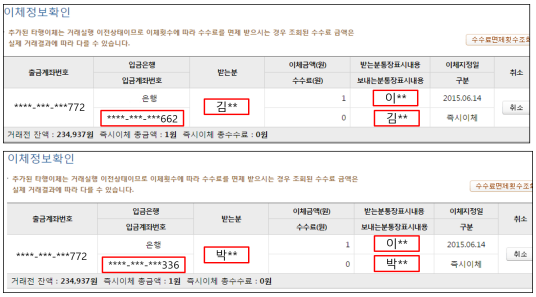

Fig. 9. Manipulated result screen of account transfer

Table 4. Table information in OTP/certificate insert html and transfer result html

Table No.	Value
2	Withdrawal Account No.
1	Deposit Bank
6	**Deposit Account No.**
2	**Receiver**
3	Transfer Amount(Won)
7	Fee(Won)
4	Recipient's Bankbook Indication Details
8	**Displayed Information on My Account**

Table 5. result html value information

Value	Information
이**	user name
박**	original recipient name
김**	attacker recipient name
****-***-***772	user account number
****-***-***336	original recipient account number
****-***-***662	attacker recipient account number

되지만, 실제로는 공격자가 변조한 계좌번호로 입금이 완료된다. Table 5. 는 Fig. 7. 과 Fig. 9.에 강조된 부분을 설명한다.

III. 관련 연구

MITB(Man-In-The-Browser) 취약점을 분석한 [9] 논문은 자바스크립트 삽입 공격을 이용한 이체금액 변조와 자주 이체한 계좌목록을 이용한 입금계좌번호 변경 기법을 수행했다. 자바스크립트 조작으로 사용자가 입력한 입금계좌번호를 취소하고 이체계좌목록에 있는 다른 계좌번호로 변경하는 것이 가능함을 보였다. 사용자가 입력한 입금계좌번호는 공격자의 프로그램에 의해서 새로 생긴 가짜 입력 폼에 입력되게 된다. 그리고 사용자가 자주 쓰는 입금계좌번호 목록에 공격자의 계좌번호를 추가하여 자주 쓰는 입금계좌번호 목록에서 공격자의 계좌번호를 불러와서 거래를 변조하는 방식이다. 이 방식의 단점은 공격자가 원하는 계좌번호로 이체시키기 위해서는 사용자의 자주 쓰는 계좌번호목록에 공격자의 계좌번호를 추가하여야 한다는 제약이다. 하지만, 해당 논문에서는 어떻게 이를 가능하게 하는지에 대한 언급은 없었다. 또한 해당 논문에서는 입금계좌번호가 키보드 보안으로 암호화되어 있기 때문에 스크립트 수준에서는 조작하기 어렵다고 하였다. 그러나 본 논문에서 제안한 방법은 스크립트 수준에서 입금계좌번호의 변조가 가능함을 보였다.

[10] 논문은 웹 브라우저에서 사용자가 입력한 입력값의 변조를 방지하는 연구를 수행했다. 키보드로부터 입력된 값과 DOM에 저장된 값을 비교하여 변조 여부를 확인하는 방식이다. 사용자가 이체 번호를 입력할 때 KEYPRESS()함수가 호출되고, 이 함수가 호출될 때 DOM에 키 입력값을 배열로 저장한다. 이 방식은 DOM에 저장된 값의 변조를 확인할 수 있다. 하지만 사용자가 키보드로 입력한 값의 변조 공격에는 대응할 수 없다. 자바스크립트를 이용하면 사용자가 웹 브라우저에 키보드로 입력한 키 값도 변조 가능하다. 공격자가 키보드 입력값과 DOM 값을 모두 변조하면 비교 확인방법을 우회할 수 있다. 또한 키보드 입력값을 DOM에 암호화하여 저장하여도 스크립트 형태로 작동하기 때문에, 본 논문에서 사용한 공격처럼 스크립트 변조를 통하여 암호값 획득 후 이체정보 변조가 가능하다.

IV. 토 론

본 논문에서 제안한 이체정보 변조 자바스크립트 코드와 HTML 값 변경 코드는 별도의 프로그램으로 자동화되어 동작하지 않는다. BHO(Browser Help Object)를 이용하거나, 웹 브라우저 프로세서에 Fig. 10. 과 같이 dll 인젝션 공격을 이용하여 악성 dll 파일을 삽입하여 해당 공격을 자동화할 수 있다.

우리가 제안한 방식은 사용자가 입력한 번호의 암호값을 획득한 후, 우리가 원하는 공격의 계좌번호의 암호값으로 재조합하는 방식이다. 사용자가 입력한 암호값의 숫자들의 범위 내에서만 계좌번호의 암호값을 조합할 수 있다. 예를 들어, 10012935의 이체번호가 있으면 우리가 조합할 수 있는 번호는 0, 1, 2, 3, 5, 9이다. 이 6개의 번호를 사용하는 계좌번호를 사용하여야 한다. 그러나 공격자가 피싱 공격에 사용가능한 계좌번호를 충분히 확보하면, 사용자가 입력한 계좌번호에 맞춰서 공격에 사용할 계좌번호로 재조합하여 피싱 공격에 사용할 수 있다.

제안 방법은 메모리 해킹 공격과 마찬가지로 정상 인터넷 뱅킹 사이트에 접속하고 보안 모듈이 설치되어 있더라도, 키보드 보안 모듈에 의한 이체번호 암호값을 변조할 수 있다. 하지만 자바스크립트를 이용한 계좌번호 변조의 한계도 있다. 암호화 자바스크립트 문서가 난독화 되어 있으면, 난독화 된 자바스크립트 문서를 해독해야 하는 어려움이 있다.

메모리 해킹 피싱 공격은 현재 제공되는 보안 모듈을 우회하고, 자바 스크립트 코드를 삽입하지도 않음과 동시에 메모리에 저장되어 있는 이체정보를 직접적으로 수정하기 때문에 더욱 강력한 공격이다. [10]논문이 제안한 방식에서, 사용자가 입력한 값을 비교한다고 해도, 메모리 해킹 공격을 이용하면 DOM에 저장된 값을 찾아내서 변조할 수 있고, 사용자가 입력한 값 또한 변조할 수 있다. 메모리 해킹 공격의 경우에는 해당 웹 브라우저의 프로세스에 할당된 메모리를 검색하여 원하는 값의 메모리 주소 번지를 찾을 수 있다. 그 후, 그 주소 번지에 변조하고자 하는 값을 덮어쓰기로 저장하면 된다. Fig. 11. 은 우리가 구현한 메모리 해킹 공격 프로토타입으로 인터넷 뱅킹 이체시에 사용자가 입력한 이체 번호를 찾아내어 변조 가능함을 보여준다. 사용자가 입력한 값인 0123456789가 저장된 메모리의 주소 번지를 찾아내어 9876543210으로 변조하였다.

현재 국내 은행에서 피싱 공격에 대응하는 보안 서비스는 Table 6. 과 같다. 은행들이 제공하는 보안 서비스 중에서 이체정보 변조를 사전에 방지할 수 있는 서비스는 신입금계좌지정와 자금이체전화승인 서비스이다. 신입금계좌지정 서비스의 경우에는 사용자가 사전에 지정한 계좌로만 입금이 가능하도록 제공하는 서비스가 있다. 하지만 사용자가 서비스에 가입하는 것은 선택사항이기 때문에 사용자가 해당 서비스에 가입하기 전에는 본 논문에서 제안한 이체정보 변조 공격에 대응할 수 없다. 그리고 자금이체전화승인서비스는 ARS로 수신계좌와 수신인, 금액 등을 재확인하기 때문에 이체정보 변조 공격에 대응할 수 있다. 하지만 이체정보 확인을 위해 사용자의 주의가 필요하다는 점과 1일 누적 300만원 이상 자금이체시에만 동작한다는 점에서 한계가 있다. 이 서비스도 신입금계좌지정 서비스와 마찬가지로 선택사항이다. 또한 사후에 이체정보 변조 공격 여부를 확인할 수 있는 입출금 거래 내역 알림 서비스가 있다. 하지만 이미 공격자에 의해서 이체정보가 변조된 거래가 완료된 후이기 때문에 사용자는 금전적 피해를

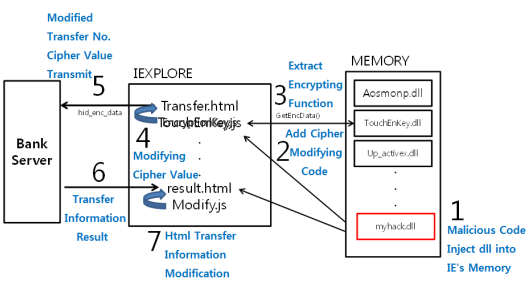

Fig. 10. Structure of dll injection for manipulation

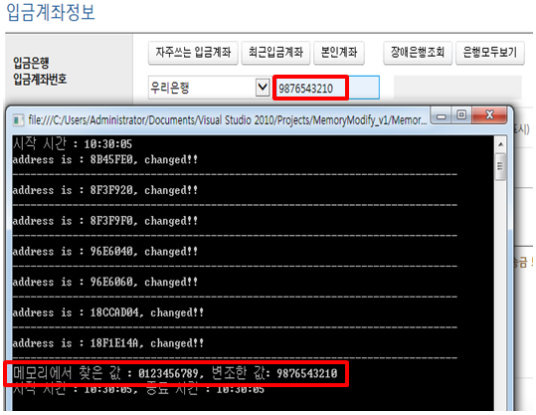

Fig. 11. memory hacking prototype

막을 수 없다.

V. 결 론

본 논문에서는 국내 주요 은행의 인터넷 뱅킹 보안 현황에 대해서 알아보고 그 중에서도 이체시에 키보드 입력 보안을 담당하는 키보드 보안 모듈의 암호화 과정을 분석하였다. 사용자가 입력한 이체정보는 암호화를 담당하는 자바스크립트에서 암호화를 하였고, 본 논문에서 제시한 이체정보 변조 코드를 자바스크립트에 삽입하여 암호값을 분석하고 재조합하였다. 이체정보는 암호화되어 메모리에 저장되어 있지만, 스크립트 수준에서 암호값을 변조하는 것이 가능하였다. 또한 사용자가 정상 인터넷 뱅킹 사이트에 접속하여 보안 모듈이 작동하였음에도 불구하고 이체정보를 변조하는 자바스크립트로 원하는 입금계좌번호의 암호값으로 변조하고 이체정보 확인 페이지도 수정하여 실제 피싱 공격이 가능함을 보였다.

본 논문에서 제안한 방식은 난독화된 자바스크립트에 대한 한계가 있지만, 향후 메모리 해킹 공격을 통해서 메모리에서 이체정보 암호값을 직접 변조 가능하도록 연구할 예정이다.

References

[1] The Bank of Korea, "Use of Internet Baning Services during 2014," Feb. 2015.

[2] Financial News, "phishing attack status of Korean banks," http://www.fnnews.com/news/201502161501395040, Feb. 2015.

[3] Seung-hyun Kim, Dae-sun Choi, Seung-hun Jin, Sung-hoon Lee, "Geo-location based QR-Code authentication scheme to defeat active real-time phishing attack," In Proceedings of the 2013 ACM workshop on Digital identity management, pp. 51-62, Nov. 2013.

[4] Han-wook Lee and Hyu-keun Shin, "Study on strong authentication to defeat memory hacking attack," Korea Institute of Information Security and Cryptology, 23(6), pp. 67-75, Dec. 2013.

[5] SoftCamp, "Secure KeyStroke 4.0," http://www.softcamp.co.kr/, Jun. 2015.

[6] Raon Secure, "TouchEnKey," https://www.raonsecure.com, Jun. 2015.

[7] AhnLab, "Ahnlab Online Security 2.0," http://www.ahnlab.com/kr/site/product/productView.do?prodSeq=34, Jun. 2015.

[8] w3schools.com, "HTML DOM innerHTML Property," http://www.w3schools.com/jsref/prop_html_innerhtml.asp, Jun. 2015.

[9] Young-jae Maeng, Dong-oh Shin, Sung-ho Kim, Dae-hun Nyang, Mun-Kyu Lee, "A vulnerability analysis of MITB in online banking transactions in Korea," Internet and Information Security, 1(2), pp. 101-118, Nov. 2010.

[10] Chang-hun Yu and Jong-sub Moon, "A study on protecting for forgery modification of user-input on webpage," Journal of The Korea Institute of Information Security & Cryptology, 24(4), pp. 635-643, Aug. 2014.

Table 6. Security service list of Korean banks

K = KB, W = Woori, N = Nonghyup, S = SC, H = Hana, I = ShinHan

Service Name	Korean Bank	Service Description
Designating PC Service	K, N	Transaction occur On Pre-Designated PC Only
Auto Notifying Service	K, W, N, S, H	Notify Transaction list to the Cell Phone in Real-time.(Balance, Transaction Amount, Amount Threshold can be set by User.)
Securiy SMS Service	K	Important Transaction History is sent to Cell Phone.
Internet Banking Call-Approval Service	K	When using Internet Banking, Bank Call to Pre-designated number and ask the decision to user directly.
Certificate in Cell Phone Service	K, N, H, I	Save Certificate on the Cell Phone and bring it back to the PC when it needed.
Designating Phone Service	K, S, H, I	Only pre-designated Phone can be used to do Banking.
Anti-phishing personal image	K	User-selected Image, personal identifying character, Color of the image is shown on the top left of the site to check that this site is legitimate or phishing site.
OTP Generator	K, W, S, H	Generate One-Time-Password using Hardware OTP Generator and using it when transaction start.
New Deposit Account Designating Service	K	Only pre-designated account can be used to deposit.
Additional Authentication Service on Important Transaction	W	Additional Authentication will be required when pre-selected important transaction(Certificate Reissuing, More than 3,000,000 Won Transfer for one day) is occurred.
Transfer Call-Approve Service	K, W,	On the final stage of the Transfer, Additional authentication through pre-designated phone number is needed.
overseas IP Ban Service	W	Transfer and Transaction wil be blocked when access with overseas IP.
2 Channel Authentication	W, H	When Logging in, additional authentication proceed using Pre-designated phone number.
Changing Account Password Service	W	Periodically recommend changing Password.
Hiding Account Service	W, N	When Internet Banking, Hide the Account from Account list.
One-Touch Remote Control Service	W	Accept or Deny Transaction using Pre-designated Smart phone.
Excepted-PC Loging Notification Service	I	Notifying Service using SMS When Unusual PC access.
Overseas IP Log in Notification Service	I	Notifying Service using Internet Banking Pop'up When accessed from oversea.
Anti-Phishing Service	K	Anti-Web Phishing Service using Security Image and identification Words.
Account Security Management	N	The Service that manage Security Account, Hiding account, Account Inquiry Prohibition Services.

〈저 자 소 개〉

이 성 훈 (Sung-hoon Lee) 학생회원
2011년 8월: 충주대학교 컴퓨터공학과 졸업
2011년 9월~현재: 과학기술대학원대학교 정보보호공학과 통합과정
2011년 9월~현재: 한국전자통신연구원 UST연구생
〈관심분야〉 개인정보보호, 인터넷 피싱, PKI, 인증

김 승 현 (Seung-hyun Kim) 정회원
2002년 2월: 금오공과대학교 컴퓨터공학과 졸업
2004년 2월: 포항공과대학교 컴퓨터공학과 석사
2004년 1월~현재: 한국전자통신연구원
〈관심분야〉 ID 관리, 모바일 지불결제, 정보보호

정 의 엽 (Eui-yeob Jeong) 학생회원
2015년 2월: 시립대학교 컴퓨터공학과 졸업
2015년 3월~현재: 과학기술연합대학원대학교 정보보호공학과 통합과정
2015년 3월~현재: 한국전자통신연구원 UST연구생
〈관심분야〉 정보보호, 모바일지불결제, 인증

최 대 선 (Daeseon Choi) 정회원
1995년: 동국대학교 컴퓨터공학과 졸업
1997년: 포항공과대학교 컴퓨터공학과 석사
2009년: 한국과학기술원 전산학과 박사
1997년~1999년: 현대정보기술
1999년~현재: 한국전자통신연구원 인증기술연구실 실장/책임연구원
〈관심분야〉 인증, 개인정보보호, 빅데이터 분석

진 승 헌 (Seung-Hun Jin) 종신회원
1993년: 숭실대학교 전자계산학과 졸업
1995년: 숭실대학교 전자계산학과 석사
2004년: 충남대학교 컴퓨터과학과 박사
1994년~1996년: 대우통신
1996년~1999년: 삼성전자
1999년~현재: 한국전자통신연구원 사이버보안기반연구부 부장/책임연구원
〈관심분야〉 컴퓨터/네트워크 보안, PKI, ID관리, 개인정보보호, 모바일 지불결제 보안