情報保護學會論文誌
第15卷 第5號, 2005. 10

URL 스푸핑을 이용한 피싱 공격의 방어에 관한 연구*

민 동 옥,† 손 태 식, 문 종 섭‡
고려대학교 정보보호기술연구센터

A Study on the Phishing Attack Protection using URL Spoofing*

Dong-og Min,† Tae-shik Shon, Jong-sub Moon‡
Center for Information Security Technologies, Korea University

요 약

최근 증가하고 있는 피싱 공격은 사용자의 실수를 유발해 개인정보를 공격자에게 노출시켜 공격자가 경제적 이득을 취하는 공격이다. 유사 메일이나 유사 주소로 사용자를 속이던 이 기법은, 메일주소 위조, 도메인 주소 위조 등으로 점점 다양해지며 기술적으로 발전해 왔다. 최근에 이르러서는 몇몇 웹 브라우저에서 발생한 취약점, 정상적인 스크립트, HTML, DNS 스니핑 등을 이용한 URL 스푸핑 공격을 피싱 공격에 이용하면서 그 피해가 크게 늘고 있다. 본 논문에서는 피싱 공격에 사용되는 고도화된 기법인 URL 스푸핑을 이용한 피싱 공격에 대해서 논의하고 이에 대한 검사방법과 예방책, 더 나아가 피싱 공격을 근본적으로 제한할 수 있는 스킴을 제안한다.

ABSTRACT

There has recently been an increase of phishing attacks, attacks which lure users into revealing their personal information to an attacker who in turn exploits this information for economic gain. The conventional methods of fooling the user with similarly modified mail or address are constantly evolving and have diversified to include the forgery of mail or domain addresses. Recently the injury incurred by these attacks has greatly increased as attackers exploit the weaknesses found on a few web browsers and used these to conduct phishing attacks based on URL spoofing. Furthermore we are now witnessing the entrance of highly advanced phishing techniques that no longer simply rely on vulnerabilities, but employ ordinary script, HTML, DNS sniffing, and the list goes on. In this paper we first discuss means of investigating and preventing the advanced URL spoofing techniques used in phishing attacks, and then propose a scheme for fundamentally restricting them altogether.

Keywords : *Phishing, URL Spoofing*

Ⅰ. 서 론

Uniform Resource Location(URL) 스푸핑(Spoofing)이란 사용자의 인터넷 웹 브라우저에 표기되어 있는 주소를 속여 특정 악성 사이트를 정상적인 사이트인 것처럼 방문하여 이용할 수 있도록 하는 변조공격의 일종이다. 즉, 사용자가 www.normal.com이라는 사이트를 이용하고 있다고 생각하게 하고, 실제로는 www.malicious.com의 사이트를 이용하게 하는 공격이다. 현재 여러 웹 브라우저의 오류나 취약점, 혹은 네트워크 계층에서의 스니핑(sniffing)을 이용하여 이러한 공격이 가능하

접수일 : 2005년 4월 19일 ; 채택일 : 2005년 8월 17일
* 본 연구는 정보통신부 대학 IT연구센터 육성·지원 사업의 연구결과로 수행되었습니다.
† 주저자, eieshine@korea.ac.kr
‡ 교신저자, jsmoon@korea.ac.kr

다. 피싱(phishing)이란 사회 공학적(social engineering)기법을 이용하여 온라인상에서 계좌번호, 카드번호, 계좌 비밀번호 등의 금융정보를 유출하여 온라인상의 금융사기를 일으키는 신종 사기 기법이다. 일반적으로 인터넷을 이용하여 금융거래를 하는 사용자를 공격대상으로 하고 있으며, 온라인 쇼핑몰 혹은, 인터넷 은행등을 사칭한 전자우편을 통해 공격한다. 초기에 사용된 피싱 공격은 전자우편 주소나 도메인 이름을 유사하게 가장하여 사용자가 착각하게끔 하는 공격이 대부분이었으나, 현재 공격 기법이 점차 고도화되어감에 따라 전자우편 주소 등을 위조하여 사용자를 완전히 속이는 기법들로 발전하였다.

URL 스푸핑 공격이나 피싱 공격 그 각각은 시스템의 운영이나 동작에 영향을 미치지 않고, 사용자의 주의를 통해 어느 정도 방어할 수 있다. 그러나, URL 스푸핑과 피싱이 결합된 형태에서는 사용자가 자신이 피싱 공격을 당하고 있다는 사실을 알아채기 힘들기 때문에, 금융사기의 피해자가 되기 쉽다. 피싱을 통한 금융사기는 한 번의 공격으로도 사용자에게 직접적인 경제적 피해를 끼칠 정도로 위험도가 높고, 공격을 당한 후 피해를 인지하는 시간이 길게는 수일씩 걸릴 정도로 공격인지도가 낮기 때문에 일반적인 웜이나 바이러스와 같이 공격을 당한 후 조치를 하는 방법은 무의미하다.

본 논문에서는 이러한 URL 스푸핑과 피싱이 결합된 공격방법에 대해서 연구하고, 이를 예방하는 방안에 대해서 제안하고자 한다. 2장에서는 피싱과 관련된 현재 연구현황과 배경이론, 3장에서는 URL 스푸핑을 이용한 피싱 공격기법 분석, 4장에서는 제안하는 피싱 공격의 대응방안, 5장에서는 공격방어 실험과 결과에 대해서 살펴보겠다.

II. 배경 연구

이번 장에서는 피싱 공격 방어 기법에 관련된 배경 연구로서 피싱 공격의 개념과 현황 및 피싱 공격에 있어 가장 널리 사용되고 또한 본 논문에서 초점을 맞추고 있는 URL 스푸핑에 대해서 알아볼 것이다.

2.1 피싱 공격의 개념과 현황

피싱은 1996년 American Online(AOL instance Messenger)를 사용하는 해커들이 일반 사용자에게 위조된 전자우편을 보내는 해킹기법에서 유래되었으며, 이 해커들은 자신이 보낸 전자우편이 AOL에서 발송된 것처럼 속이고 사용자들의 계정정보를 빼냈다.[17] 1990년대 후반 전자상거래와 인터넷 금융거래의 빠른 보급화 물결을 타고 사용자의 개인정보가 정보에 국한되지 않고 경제적인 이득과 직결되자 해커들은 위조된 전자우편을 발송하는 기법에 착안하여 사용자의 개인정보를 빼내고자 하였다. 피싱은 주로 스팸메일(spam-mail)과 같은 수신자가 원하지 않는 전자우편을 다량으로 발송하여 합법적인 전자상거래나 인터넷 은행과 같은 금융관련 사이트로 유인하여 사용자들로 하여금 계좌정보 갱신, 로그인, 회원정보 확인 등의 이유로 비밀번호, 주민등록 번호, 계좌번호, 신용카드 번호 등을 입력하도록 요구한다. 초기 피싱 공격은 유사한 메일이나 도메인으로 접속을 유도하는 사회공학적 기법을 사용하였다. 현재 컴퓨터 기법들이 고도화 되어감에 따라 웹 브라우저의 취약점, 네트워크 장비들의 스니핑(sniffing), 혹은 스니핑을 이용한 하이재킹(hijacking)등이 피싱 공격에 사용되고 있으며, 웜 바이러스를 이용하는 기법까지 등장하였다. 컴퓨터 전문가조차 주의 깊게 살펴보지 않으면 속을 수 있는 고도화된 기법의 등장으로 현재 스푸핑을 통한 피싱 공격은 급격하게 늘어가고 있다. Anti-Phishing Working Group(APWG)의 보고에 따르면 그림 1에서 보듯이, 피싱 공격에 이용된 사이트는 점차적으로 증가하여 2004년 10월 첫 주 161건이었던 것이 인터넷 익스플로러 URL 스푸핑 취약점이 발표되면서 급격히 증가하는 추세를 보여 2005년 1월 마지막 주에 948건을 기록하였다.[1]

또한, 월간집계를 통해 2005년 1월에 집계된 피싱사이트는 2560개로 전월인 2004년 12월에 비해 28% 증가하였고, 2004년 10월에 비해 2배 이상 증가하여 점진적으로 피싱사이트가 늘어나고 있는 것을 알 수 있다. 피싱사이트 한 개당 일어나는 피

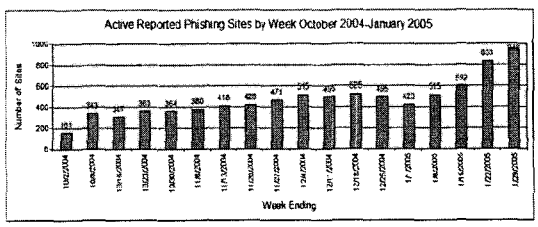


그림 1. 2004년 10월~2005년 1월 주간 피싱(Pishing) 사이트 집계 현황

싱사고의 건수가 작게는 수 십개에서 많게는 수 백개에 이르고, 접수되지 않은 피싱사이트를 감안해 보면 피싱 사고는 피해규모는 더욱 클 것으로 추정된다. 그럼에도 불구하고, 현재 피싱과 관련된 연구는 몇몇의 보안업체 등에서 제한적으로 이루어지고 있으며 관련자료는 대부분 보안관련 업체나 커뮤니티 그룹의 화이트페이퍼에 의존하고 있는 실정이다.

2.2 URL 스푸핑 기법

서론에서 언급한대로 URL 스푸핑은 웹 브라우저의 취약점이나 스크립트를 이용하는 악성코드를 사용하여, 사용자에게 방문중인 사이트가 아닌 다른 사이트의 주소를 보여주어 사용자를 혼란시킨다. 현재 알려진 기법들로는 특수문자를 사용한 변조, 스크립트를 사용한 변조, HTML 태그를 이용한 변조, ISA Server의 Reverse Lookup 취약점, 네트워크 스니핑을 통한 도메인 네임 스푸핑, 악성코드를 통한 정보유출과 hosts파일 위/변조 등이 있다.[3]

특수문자를 사용한 변조기법은 마이크로소프트의 인터넷 익스플로러 특정버전에 포함된 취약점을 이용하는 기법으로, 페이지를 지정하는 URL에 특수문자를 포함시켜 주소 표시줄을 지우고 다시 표시하는 기법이다. 이 취약점은 인터넷 익스플로러의 URL을 표준형태로 변환하는 코드에서의 오류로 인해 발생한다.[7]

스크립트를 사용한 변조기법은 특정버전의 웹 브라우저에서 스크립트의 window.open 함수를 이용하여, 새 창으로 열린 웹 브라우저의 주소를 공격자가 원하는 사이트로 표시하도록 하는 기법이다.[9] 공격자는 이것을 이용하여 사용자가 주소표시줄의 주소와 다른 위조된 사이트로 접근을 하도록 유도할 수 있다. 또한, 주소가 표시되지 않는 작은 창을 정상적인 사이트와 함께 보여주어, 실제 사이트에서 제공하는 창이라고 믿게끔 한다.[13] 보통 이러한 창에는 로그인 관련된 정보나, 개인정보를 변경하라는 내용의 문구가 실려 사용자의 실수를 유도한다.

HTML 태그를 이용한 변조기법은 웹 사이트를 꾸미는 가장 기본적인 Hypher Text Markup Language(HTML)의 〈IFRAME〉,[10] 〈TABLE〉,[11] 〈A HREF〉 태그들의 특징을 이용하여 웹 브라우저의 상태바나 주소를 위조하는 기법이다. 이러한 방법은 특정 웹 브라우저의 취약점을 이용한 변조 공격이 아니기 때문에, 웹 브라우저의 버전이나 종류에 관계없이 변조가 가능하다.

ISA서버의 Reverse Lookup 취약점을 이용한 기법은, ISA서버와 같이 인터넷과 내부 네트워크의 인터넷 게이트웨이 역할을 하며, 웹 캐시 기능을 하는 서버의 Reverse Lookup[1]의 결과를 캐시에 저장하는 것을 이용한 기법이다.[3] 공격자들은 이점을 이용해 위조된 도메인이름과 IP를 가진 Reverse Lookup 응답 메시지를 ISA서버의 웹 캐시에 주기적으로 보내어 캐시를 구성하도록 한다.

네트워크 스니핑을 이용한 도메인네임 스푸핑 기법은 Domain Name System Security Extension(DNSSEC)[4]을 사용하지 않는 네트워크에서 스니핑 툴 등을 이용하여 사용자의 DNS질의를 가로채어 사용자에게 다른 IP를 반환하는 기법이다. 공격자는 네트워크를 스니핑하여 질의나 패킷 등을 관찰하다가 사용자가 DNS질의를 요청하였을 때, 사용자에게 공격자가 변조한 사이트의 IP를 반환한다. 사용자의 PC에는 정상적인 사이트로 표시되지만, 공격자가 변조한 사이트에 접속하게 된다.

어떠한 악성코드는 감염된 시스템의 hosts 파일을 변조하여 해당 사이트로의 접근을 막는 기능을 한다. 지금까지 발표된 악성코드는 대부분 hosts 파일에서 안티-바이러스 사이트 등 접근을 차단할 사이트들을 127.0.0.1로 IP를 할당하여 자기귀환(Loop-back)시켜 해당 사이트로의 접근을 하지 못하도록 한다. 이것은 웹브라우저가 주소를 찾으라는 명령을 받았을 때, DNS에 질의를 보내기에 앞서 시스템의 hosts파일을 검사하여 IP를 반환하기 때문에 가능하다. hosts파일의 위/변조를 통한 스푸핑 기법은 공격자들이 이점을 이용해 원하는 금융관련 사이트의 도메인을 위조한 IP로 대응하게끔 hosts파일을 변조하는 악성코드를 배포하고, 악성코드에 감염된 사용자가 공격자가 설정해 놓은 사이트에 접속하도록 유도하는 방법이다. 악성코드 감염자의 웹 브라우저는 정상적인 주소를 표시하면서 위조된 사이트에 접속하게 된다.

Ⅲ. URL 스푸핑을 이용한 피싱 공격 분석

3.1 URL 스푸핑을 이용한 피싱 공격방법

URL 스푸핑을 이용한 피싱 공격은 사회공학적

1) 역방향조회. IP로 도메인네임을 조회

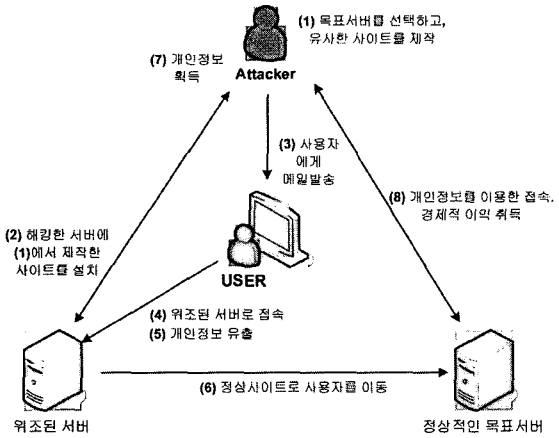


그림 2. 스푸핑을 이용한 피싱 공격 과정

공격기법과 기술적인 기법이 병합되어 있는 방법으로 방어하기가 까다로운 기법이다. 일반적인 피싱 공격 과정과 유사하나, 사용자에게 스푸핑 된 URL을 보여주는 악성코드가 들어있는 메일을 발송한다는 점이 일반적인 피싱 공격과는 다르다.[17] URL 스푸핑을 이용하여 피싱 공격을 하는 일반적인 과정은 다음의 그림 1에 나타나 있다. 그림 속의 숫자는 과정의 순서를 나타낸다.

(1) 공격자 : 유사 사이트 제작
(2) 공격자 -〉 위조된 서버 : 유사 사이트 설치
(3) 공격자 -〉 사용자 : 위조된 사이트로 유도하는 악성 메일
(4) 사용자 : 위조된 사이트로 접속
(5) 사용자 -〉 위조된 사이트 : 개인정보 유출
(6) 사용자 : 위조된 사이트 -〉 사용자 : 정상 사이트
(7) 위조된 사이트 -〉 공격자 : 사용자의 개인정보
(8) 공격자 -〉 정상 사이트 : 사용자를 가장하여 접속

공격자는 피싱 공격에 이용할 목표사이트를 선정하고, 유사사이트를 제작한다. 그리고, 제 3의 서버를 해킹하여 유사사이트를 설치하고, URL을 스푸핑하는 코드가 담겨진 악성 메일을 발송하여 피해자의 접속을 유도한다. 피해자가 위조된 사이트로 접속하여 로그인 하거나, 혹은 개인정보 변경화면을 통해 개인정보를 변경하면 이 정보는 공격자에게 전해지고, 피해자는 다시 정상적인 사이트로 이동된다. 공격자는 피해자에게 받은 개인정보 혹은 계정과 패스워드를 통해 정상적인 사이트에 접속하고, 이를 이용해 이득을 취한다.

3.2 알려진 대응방안

현재 알려진 피싱 공격의 대응방안은 웹 브라우저의 취약점을 패치하고, 사용자가 사이트의 주소 표시줄을 확인하는 수동적인 방법이 전부이다.[5] 그러나, 피싱 공격의 대상이 임의의 사용자이고 주로 피해를 입는 피해자는 피싱 공격에 무지하거나, 혹은 컴퓨터에 많은 지식이 없는 사용자들이다. 그렇기 때문에 사용자가 직접 사이트의 주소를 확인해서 위조된 사이트인지를 판별하는 것이 피싱 공격을 방어하는 근본적인 해결책이 되지는 않는다. 게다가, 접속하는 웹 사이트마다 모두 주소를 확인하고 피싱 여부를 검사하는 방법도 다소 무리가 있으며, ISA 서버로의 Reverse Lookup이나 DNS, hosts파일 변조를 통한 피싱 공격은 주소 확인을 통해서 위조된 사이트의 판별이 불가능하다. 이 절에서는 알려진 피싱 공격의 대응방안과 한계에 대해서 간략하게 서술하도록 하겠다.

3.2.1 최신패치의 적용

웹 브라우저의 취약점으로 인해 URL 스푸핑이 가능한 공격들은 벤더에서 제공한 최신 패치를 통해 막을 수 있다. 특히, 일반적으로 가장 폭넓게 사용하고 있는 마이크로 소프트의 인터넷 익스플로러의 경우 윈도우즈 XP 서비스팩 2버전이 아닌 IE 5.0, 5.5, 6.0등 모든 버전에서 취약점이 노출되어있다. 꼭 인터넷 익스플로러만이 아닌 공개용 웹 브라우저에도 유사 취약점이 존재하기 때문에 패치를 해야한다.

그러나, 패치를 적용하여 웹 브라우저의 취약점이 없어진다고 하더라도, 스크립트나 HTML태그를 이용한 정상적인 기술로 위조된 사이트에는 대응할 수가 없다. 그렇기 때문에 최신패치의 적용은 피싱방지의 부가적인 대응수단의 하나일 뿐 피싱 공격의 대응책은 될 수 없다.

3.2.2 스크립트를 이용한 확인

주소 표시줄에 다음과 같은 스크립트를 입력하여 실행시킨다. 해당 스크립트는 현재 방문하고 있는 웹 페이지의 실제 URL 정보를 보여주며, 이것을 방문하고 있는 웹 페이지의 URL과 비교한다.

- javascript:alert("Actual URL address:"+ location.protocol+"//"+locationhostname+ "/");

다음 코드는 위의 코드를 이용하여 현재 방문하고 있는 URL과 실제 URL을 모두 보여서 비교하게 해준다.

```
·javascript:alert("The actual URL is:
\t\t"+location.protocol+"//"+location.
hostname+"/"+"\nThe address URL is:
\t\t"+location.href+"\n"+"\nIf the server
names do not match, this may be a
spoof.");
```

3.2.3 인터넷 익스플로러의 열어본 목록 창을 통한 확인

인터넷 익스플로러에서 제공하는 열어본 목록 창은 지금가지 방문한 웹 페이지의 URL이 기록되어 있다. 이 창을 실행하면 현재 방문하고 있는 웹 페이지의 URL이 보이는데, 이 URL과 방문하고 있는 웹 페이지의 URL을 비교하여 검사할 수 있다.

3.2.4 웹 브라우저의 새 창에서 URL 확인

대부분의 스푸핑은 방문하고 있는 웹 페이지의 내용을 속이며, 주소표시줄에 다른 URL을 보여주는 기법을 사용한다. 이런 경우 새로운 웹 브라우저를 실행시켜 방문하고 있는 웹 페이지의 URL로 접속하여 새로운 세션을 맺게 되면, 위조된 사이트가 아닌 실제 사이트에 접속할 수 있다. 그러나, 전자상거래 사이트와 같이 세션이나 쿠키를 사용하는 사이트에서는 세션을 새로 맺었기 때문에 사용에 제약을 받을 수 있다.

3.2.5 속성을 이용한 확인

인터넷 익스플로러의 경우 방문하고 있는 사이트의 등록정보(속성)을 이용하여 실제 접속중인 URL과 주소표시줄에 표시된 URL을 비교할 수 있다. 인터넷 익스플로러뿐 아니라 다른 웹 브라우저도 이런 기능을 지원하고 있다.

3.2.6 하이퍼링크가 사용하는 URL을 확인

수신된 이메일에 있는 하이퍼링크를 인터넷 익스플로러 기능중의 하나인 "바로 가기 복사"를 이용해 메모장과 같은 문서편집기에 복사하여 URL을 확인한다. 이 방식으로 3.1에서 언급한 특수문자를 사용한 주소 변조를 판별해 낼 수 있다. 복사된 URL에 %00, %01, @과 같은 특수문자가 포함되어 있으면, 그 하이퍼링크는 위조된 주소로 사용자를 이동시키는 연결이다.

Ⅳ. 제안하는 방안

위의 피싱 공격 대응방안을 살펴봤을 때, URL을 사용자가 직접 확인하는 방법이 가장 효율적인 방법이 될 수도 있겠다. 하지만, 이 방법은 ISA서버의 Reverse Lookup이나 DNS 질의를 가로채어 다른 응답을 보내는 방식의 공격에 대해서는 적용할 수 없다. 게다가, 전적으로 사용자의 직접적인 확인이 필요하므로, URL을 직접 확인하는 작업 할 능력이 없는 사용자는 여전히 피싱 공격의 위험에 노출되어진다.

URL 스푸핑을 이용한 피싱 공격을 해결하기 위한 가장 간단한 방법은 사이트가 위조되지 않았음을 사용자가 신뢰하는 것이다. 그러기 위해서, 사이트는 사용자에게 자신이 위조되지 않았음을 증명하여야 하며, 이를 위해 사이트는 사용자와 사이트가 모두 신뢰할 수 있는 제 3의 기관(TTP, Trust Third Party)에 인증을 받아야한다. 인증을 통한 사용자와 사이트간의 신뢰관계를 형성하여 피싱을 방지하는 프레임워크는 다음 그림 3에 나타나 있다.[20]

(1) 사용자 -> 웹 서버 : 세션 설정, 정보 요청
(2) 웹 서버 -> 사용자 : $E_{Pr\_web}(R)$, R[2]
(3) 사용자 -> TTP : $E_{pb\_ttp}(IP)$[3]

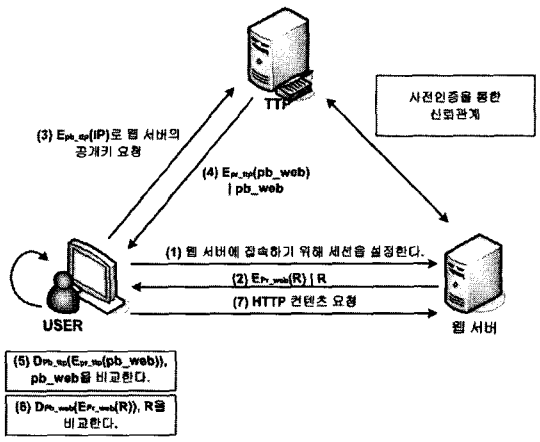


그림 3. TTP를 통해 사이트 인증을 이용한 피싱검사 모델

2) Pr_web : 웹 서버의 개인키, R : 임의의 문자열
3) Pb_ttp : TTP의 공개키

(4) TTP -> 사용자 : $E_{pr\_ttp}(Pb\_web)$, Pb_web[4]
(5) 사용자 : Compare($D_{Pb\_ttp}(E_{Pr\_ttp}(Pb\_web))$, Pb_web)[5]
(6) 사용자 : Compare($D_{Pb\_web}(E_{Pr\_web}(R))$, R)[6]
(7) 사용자 -> 웹 서버 : HTTP Request

사용자는 웹 서버에 접속을 하면서, 세션을 설정하고 웹 서버의 정보를 요청한다. 웹 서버는 사용자에게 임의의 문자 R을 웹 서버의 개인키 Pr_web로 암호화시킨 $E_{Pr_web}(R)$과 R을 함께 보내어 사용자가 증명 할 수 있도록 한다. 사용자는 신뢰할 수 있는 TTP에게 현재 세션의 IP를 TTP의 공개키 Pb_ttp로 암호화하여 전송하고, 웹 서버의 공개키를 요청한다. TTP는 웹 서버의 공개키를 TTP의 개인키 Pr_ttp로 암호화 시킨 $E_{pr_ttp}(Pb_web)$과 Pb_web을 함께 보내어 사용자가 증명할 수 있도록 한다. 사용자는 TTP의 공개키 Pb_ttp로 TTP에게 전송받은 $E_{pr_ttp}(Pb_web)$을 복호화 하여 Pb_web을 검증하고, 이를 이용해 웹 서버가 보내온 정보인 $E_{Pr_web}(R)$을 복호화한다. 복호화된 정보 $D_{Pb_web}(E_{Pr_web}(R))$와 R을 비교하여 웹 서버가 신뢰할 수 있는 사이트인지 증명한다.

위의 프레임워크는 사이트가 사용자에게 위조되지 않았음을 증명하는 방식으로, 간단하지만 매우 효과적이다. 만약 서비스를 하는 모든 사이트들이 TTP의 인증을 받아 사용자가 사이트를 증명할 수 있도록 제공한다면, 이 프레임워크는 피싱사고를 방지하는 가장 이상적인 모델이 될 수 있을 것이다. 그러나, TTP에의 인증은 서비스를 하는 웹 사이트가 자발적으로 하여야 하기 때문에, 모든 웹 사이트가 TTP에 인증을 받는다고 생각하기 어렵다. TTP에 인증을 받지 않은 사이트의 경우에는 위조여부를 판단할 수가 없기 때문에, 인증을 받지 않은 사이트의 위조를 판별할 수 있는 대책이 필요하다.

인증절차를 거치지 않은 사이트의 위조를 판단하는 간단한 방법에는 설정되어있는 세션의 IP와 현재 방문하고 있는 사이트의 IP를 검사하여 비교하는 방법이 있다. IP를 비교하였을 경우 사용자는 비교의 결과가 같으면 위조되지 않은 사이트, 다르면 위조된 사이트로 판단할 수 있다. 이 방법을 사용하기 위해서는, 현재 방문하고 있는 사이트의 IP를 질의하여 검사해야 한다. 물론, TTP에 등록되지 않은 모든 도메인을 질의하여 반환된 IP를 비교하는 방법이 가장 효과가 좋겠지만, 이것은 네트워크와 시스템의 성능에 상당한 부담을 주게 된다. 이를 해결하기 위해, 질의하는 도메인의 범위를 제한할 필요가 있으며, 프레임워크의 목적이 피싱 공격의 방어에 있으므로, 피싱 공격에 이용되는 사이트인 은행, 증권회사, 쇼핑몰 등 전자상거래가 가능한 사이트들로 한정짓는다. 논문에서는 이 사이트들을 주요사이트(Principal Web-Site)라고 명명하고, 이 주요사이트들이 구성된 목록을 주요사이트 목록 테이블(PLT : Principal Web-Site List Table)이라고 한다. 이 주요사이트들은 각 국가나 환경에 따라 다를 수 있기 때문에, 환경에 맞는 주요사이트 목록을 구성하는 것이 중요하다. PLT의 구성은 다음 표 1과 같다.

PLT검사를 통해 반환된 IP는 검사의 정확성을 위해 DNS에 한번 더 질의하며, 세션 IP, DNS 질의를 반환 값을 비교하여 그 결과를 기준으로 피싱을 검사한다. TTP의 인증을 받지 않은 사이트의 위조를 판별하는 프레임워크는 그림 4에 나타나 있다.

표 1. 주요 사이트 목록 테이블

사이트 이름	IP 주소	갱신 날짜
www.citibank.com	192.193.217.120	2004/03/15 22:00
www.daum.net	211.32.117.60	2004/03/15 22:00
www.hanabank.com	211.32.7.135	2004/03/15 22:00
www.kbstar.com	211.181.199.211	2005/01/07 16:20

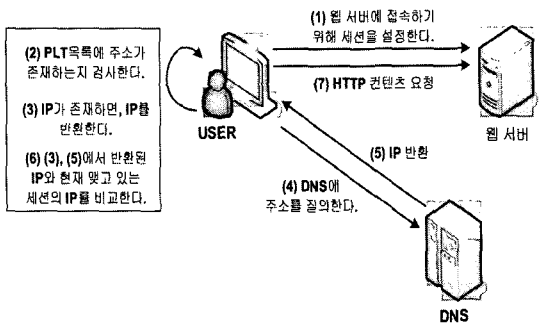


그림 4. PLT목록과 DNS질의를 이용한 피싱검사 모델

4) Pr_ttp : TTP의 개인키, Pb_web : 웹 서버의 공개키
5) Compare(DPb_TTP(EPr_TTP(Pb_web)), Pb_web) : TTP가 개인키로 암호화한 웹 서버의 공개키를 TTP의 공개키로 검증
6) Compare(DPb_web(EPr_web(R)),R) : 웹 서버가 개인키로 암호화한 R을 웹 서버의 공개키로 검증

(1) 사용자 : 세션 설정
(2) 사용자 : Search(PLT ∋ URL)[7]
(3) 사용자 : R_A(IP)[8] 또는 "없음"
(4) 사용자 -> DNS 서버 : 접속한 사이트의 URL
(5) DNS서버 -> 사용자 : R_B(IP)[9]
(6) Compare(세션 IP, R_A(IP), R_B(IP))
(7) 사용자 -> 웹 서버 : HTTP Request

사용자는 웹 서버에 접속을 하면서, 세션을 설정하고 웹 서버의 정보를 요청한다. 웹 서버로부터 인증 정보가 오지 않는다면, 사용자에게 설치된 응용프로그램은 웹 서버의 주소가 PLT목록에 존재하는지 확인한다. 존재하지 않는다면, 프로그램은 웹 서버의 주소가 PLT에 존재하지 않는다는 것을 알려주고, 존재한다면 이 IP를 반환하여 DNS에 질의한다. DNS에서 반환된 주소와 PLT에서 반환된 주소, 그리고 세션에서 설정된 IP의 주소를 비교하여 모두 같으면, 위조되지 않음으로 간주하고 사용자에게 알려준다. 만약, TTP에 인증되지 않고 PLT목록에 존재하지 않는 금융거래 관련 사이트에 방문중이라면 그 사이트는 피싱 공격을 위해 위조된 사이트일 가능성이 높다. 그리고, PLT목록에 존재하는 사이트의 각 IP 비교 결과가 정상으로 나오지 않는 경우, 그 결과를 관리자나 사용자가 설정한 보안정책에 따라 처리한다. 다음 표 2에 각 IP의 비교에 따른 결과를 정리하였다.

TTP를 통해 인증을 받지 않은 사이트의 위조를 판별하는 방법은 제외된 목록에 대하여 단순하면서도 효과적으로 피싱을 검사하고 방지하며, 시스템과 네트워크에 걸리는 부하를 최소화한다. 그러나, 이 PLT에 속한 사이트의 IP주소나 도메인 이름이 갱신되었을 때 PLT도 반드시 갱신 해 주어야 하는 문제가 발생한다. 이것을 해결하기 위해서, 사용자의 PLT를 갱신시켜주는 PLT분배 서버가 필요하다. 그리고, PLT와 DNS는 동기화되어 있어야하며 완벽하게 동기화되지 않았을 때, 동기화가 이루어질 때까지 거짓 경고를 발생하게 된다. 표 2의 "PLT≠DNS=세션" 혹은 "PLT=세션≠DNS"의 상태에 거짓 경고가 발생할 경우가 표시되어 있으며, 사용자나 관리자는 이런 거짓 경고에 대한 임계치를 책정

7) Search(PLT ∋ URL) : URL이 PLT에 포함되어 있는지 검사
8) R_A(IP) : PLT에 존재하는 URL의 IP주소
9) R_B(IP) : DNS가 사용자에게 반환한 IP주소

표 2. PLT의 반환 IP, DNS의 반환 IP, 현재 세션의 IP와의 관계와 상태

	상 태
PLT=DNS=세션	· 정상적인 접근
PLT=DNS≠세션	· 스크립트, 특수문자, HTML등을 통한 피싱 공격
PLT≠DNS=세션	· DNS, hosts파일 변조, ISA서버의 Reverse Lookup을 통한 피싱 공격 · PLT의 목록에서 IP가 갱신되지 않았음(거짓 경고)
PLT=세션≠DNS	· DNS의 IP가 갱신되지 않았음(거짓 경고)
PLT≠DNS≠세션	· 스크립트, 특수문자, HTML등을 통한 공격과 DNS, hosts파일변조, ISA 서버의 Reverse Lookup을 이용한 공격의 복합적인 피싱 공격 · PLT의 오류와 스크립트, 특수문자, HTML등을 통한 피싱 공격이 복합적으로 이루어짐

표 3. 피싱에 이용되는 URL스푸핑 기법과 대응방안

기법 / 대응방안	특수 문자	스크 립트	HTML 태그	ISA 서버 취약점	DNS 질의 하이잭	악성 코드
제안하는 방안	○	○	○	○	○	○
패치 적용	○	△	×	×	×	×
스크립트로 URL확인	○	○	○	×	×	×
열어본 목록창 확인	○	○	○	×	×	×
새 창에서 URL 확인	○	○	○	×	×	×
등록정보를 이용한확인	○	○	○	×	×	×
하이퍼링크 URL 확인	○	○	○	×	×	×
DNS재질의	○	○	○	×	×	×
DNS재질의 (DNSSEC)	○	○	○	×	○	×

해 놓는 것도 좋은 방편이 될 것이다.

다음의 표 3은 피싱 공격 대응방안을 통해 해결할 수 있는 스푸핑기법의 관계를 나타내었다. 스크립트 변조기법은 패치의 적용으로 일부 해결이 되고, 일부는 해결 할 수 없기 때문에 △로 표시하였다.

V. 실험 및 결과

5.1 실험 방안

제안된 방안의 실험을 위해 윈도우 기반에서 비주

얼 베이직 6.0을 이용하여 간단한 서핑 기능이 있는 웹 브라우저를 제작하였다. 웹 브라우저는 위에 제안한 방법에 따라 피싱을 검사하는 기능을 탑재했으며, TTP를 통한 인증에 실패하였을 경우 자동적으로 PLT검사를 하도록 했다. 실험에서 TTP는 자체적으로 응답하여 신뢰할 수 있다고 가정하였으나, 실제네트워크에 적용시에는 사용자와 TTP간의 암호화 통신을 통한 신뢰구간 확보가 반드시 필요하다. 피싱 공격의 실험은 스크립트, HTML등 웹 브라우저의 취약점을 이용한 피싱 공격과 ISA서버, DNS하이재킹 등 도메인 질의 반환값을 위조하는 피싱 공격의 검사로 나누어 실험하였고, TTP를 통해 사이트를 인증하는 방안을 시뮬레이션 하였다. 실험 결과의 확인을 수월하게 하기 위하여, 피싱 공격에 사용되는 유사사이트를 잘 알려진 국내의 포탈 사이트로 대체하였다.

웹 브라우저의 취약점을 이용한 피싱 공격 검출실험은 주소창에 특수문자를 포함시켜 URL을 스푸핑하는 기법을 사용하였다. 웹 브라우저가 www.citibank.com의 도메인을 표시하면서 www.naver.com에 접속하는 것을 보였으며, 이것을 PLT기반으로 검출하는 과정을 보였다. 검출에 사용된 PLT 목록은 그림 5에 나타나 있다.

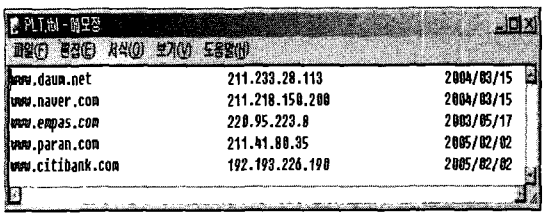


그림 5. 실험에서 사용한 PLT목록

정상적인 도메인 질의의 반환값을 위조하는 피싱 공격 검출실험은 일반적으로 도메인 질의 시, hosts 파일 검사, 웹 캐쉬 질의, DNS 질의의 순서를 거치므로, 도메인 질의 우선순위가 가장 높은 hosts 파일의 위조를 시뮬레이션 하였다. 웹 브라우저가 주소창에 www.daum.net을 표시하고, www.empas.com에 접속하는 것을 보이고, 이것을 PLT를 이용하여 정상적으로 검출하는 과정을 보였다. 위조된 hosts파일은 그림 6에 나타나 있다.

TTP를 통한 사이트의 인증을 시뮬레이션 하기 위해, 내부 네트워크에 웹 서버를 실행 시켜 환경을 만들었다. 실험결과의 확인을 수월하게 하기 위해 개인키로 서명된 암호문과 평문은 〈META〉태그에 삽입하여 전송하였고, 프로그램은 TTP에 IP로 요청하여 받아온 공개키로 서명을 확인하였다. TTP는 내부네트워크에서 소켓을 사용하여 통신하였으며, 요청 받은 IP에 대한 공개키 반환 기능만을 구현하였다.

5.2 결과 및 분석

웹 브라우저의 취약점을 이용한 피싱 공격의 검출실험 결과는 그림 7에 나타나있다.

그림 7에서 보듯이 주소창에 특수문자 %01을 포함한 주소 "http://www.citibank.com%01@www.naver.com"을 입력하여 www.naver.com에 접속하였다. TTP를 통해 인증을 받은 사이트가 아니므로, 현재 세션의 IP 211.218.150.200과 도메인을 DNS에 질의하여 반환 받은 IP 192.193.226.190 그리고, PLT에 질의하여 반환받은 192.193.226.190을 비교하여 피싱사이트 여부를 검사한다. 결과

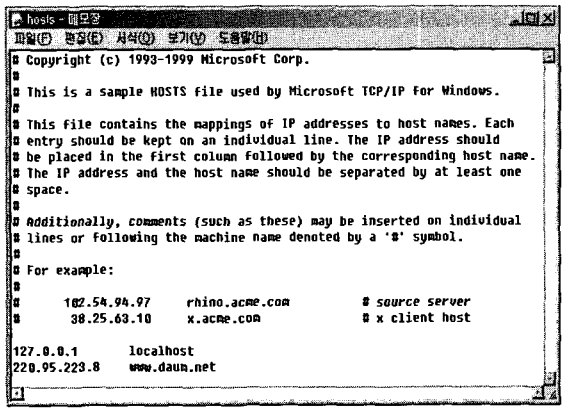


그림 6. 위조된 hosts 파일

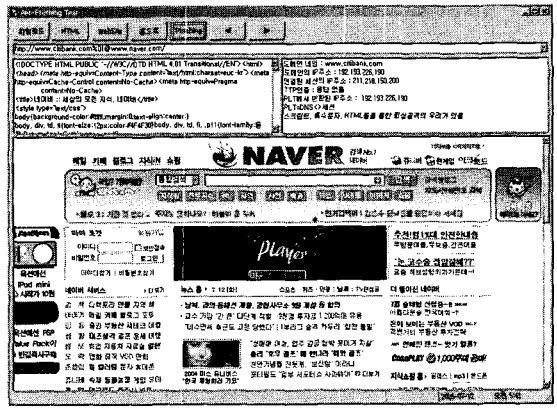


그림 7. 웹 브라우저의 취약점을 이용한 피싱 공격의 검출실험

는 표 2에 나온바와 같이 "PLT=DNS≠세션"이므로, "스크립트, 특수문자, HTML등을 통한 피싱 공격"으로 검출되었다. 그림 7의 우측 상단창에 피싱 검사의 결과가, 좌측 상단창에 현재 받아온 HTML 소스가 나타나있다.

정상적인 도메인 질의의 반환값을 위조하는 피싱 공격의 검출실험 결과는 그림 8에 나타나있다.

실험 결과, 현재 주소창에 보여지는 도메인 이름 www.daum.net 으로 질의한 IP 주소가 세션의 IP주소 220.95.223.8로 같음을 확인할 수 있다. 결과창에서 보여지듯이 도메인의 IP주소가 스푸핑되어 오기 때문에 알려진 방안들로는 검출이 불가능하고, PLT를 통해서 검출할 수 있다. 그림 5의 PLT 목록을 검사하여 확인한 IP주소는 211.233.28.113으로 세션의 IP주소와 같지 않음을 확인할 수 있다. 도메인 질의에서 반환받은 IP와 맺어진 세션의 IP가 같고, PLT목록에 있는 IP와 다르기 때문에, 표 2에서 "PLT≠DNS=세션"의 결과를 얻어 "PLT의 목록이 갱신되지 않았거나 hosts파일 등의 위조공격"으로 검출되었다.

위의 두 실험은 TTP를 통한 사이트 인증을 못했을 경우, 피싱 공격의 검출 실험이었다. TTP를 통해 사이트를 인증하는 스킴은 4장 제안하는 방안에서 언급했듯이, 사이트가 위조되지 않았음을 증명하는 스킴이기 때문에, 거짓이나 위조공격에 대한 실험은 하지 않고, TTP를 통해 사이트를 인증하는 과정만을 시뮬레이션 한다. TTP를 통한 사이트 인증 프레임워크의 시뮬레이션 결과는 그림 9에 나타나 있다.

그림 9의 좌측 상단의 HTML 소스를 확인하면 사이트 인증을 위한 암호문과 평문이 들어있음을 볼 수 있다. 내부 네트워크의 실험환경이라 도메인 네임은 IP가 그대로 보여졌으며, TTP에 IP를 질의하여 반환받은 공개키 값 103으로 암호문 "03106490 89013400317972092177306856"을 복호화하여 사이트가 전송해 온 평문과 같은 평문 "DKENWBQU"임을 확인할 수 있다. 공개키 서명 알고리즘은 RSA를 이용하였다.

실험을 통해 TTP를 통한 사이트의 인증으로 피싱 공격을 사전에 방지할 수 있고, 사이트가 인증되지 않는 경우 PLT를 이용하여 피싱 공격을 효과적으로 검출할 수 있음을 보였다. 그러나 TTP를 통한 사이트 인증 프레임워크에서 사이트 인증을 위한 과부하 실험이나, TTP에서 제공하는 서버의 과부하 실험등은 실험환경의 미비로 이루어지지 않았다. 하지만, 사이트에서 제공하는 정보나, TTP에서 제공하는 정보의 크기가 작고, 프로토콜이 단순화되어있는 것을 감안하면 피싱방지 프레임워크로 인해 발생하는 과부하는 크게 문제되지 않을 것으로 보인다.

Ⅵ. 결 론

전자우편, 하이퍼링크, DNS 하이재킹, ISA Reverse Lookup등의 인터넷 금융사기 공격은 모두 피해자가 자발적으로 개인정보를 유출하는 환경을 만든다는 공통점을 가지고 있다. 이러한 기법들은 주의를 통한 예방이 까다롭고, 공격대상이 보안에 둔감한 일반 사용자들이기 때문에 심각한 문제가 되고 있다. 게다가, 이 문제를 해결할 방안도 아직 미흡한 실정이다.

본 논문에서 제안한 피싱 검사 모델은 사용자의 간섭이 필요 없는 자동화된 모델로써, 인증을 통해

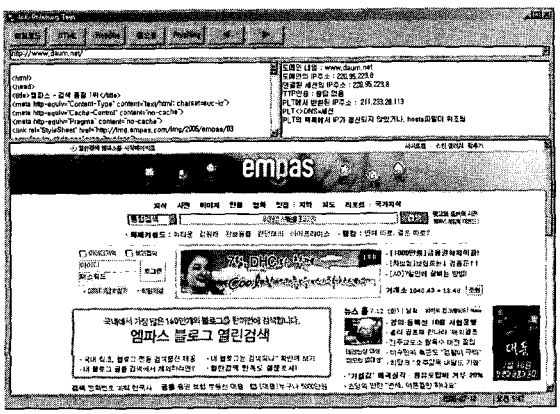


그림 8. 도메인 질의 반환값을 위조하는 피싱 공격의 검출실험

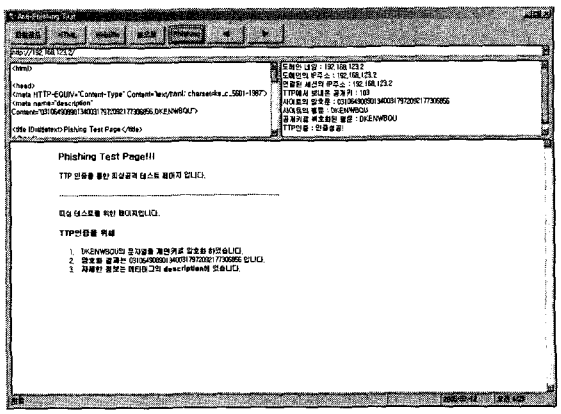


그림 9. TTP를 통한 사이트 인증 프레임워크

사용자와 사이트와의 신뢰관계를 형성함으로써 근본적으로 피싱 공격을 사전에 방지할 수 있다. 또한, Pharming과 같은 피싱과 다른 형태의 공격이 발생하였을 경우에도, TTP의 방어나 PLT 갱신을 통해 해당 도메인으로의 접속을 제한하여 피해를 예방할 수 있다. 그러나, 제안한 검사모델은 새로운 세션을 통한 접근에 초점이 맞추어져 있기 때문에, 전자우편에 개인정보를 입력하는 것을 유도하는 등의 단순화된 사회공학적 기법을 통한 피싱 공격은 방어하기가 까다롭다. 이런 피싱 공격은 사용자의 주의를 통하여 쉽게 예방할 수 있지만, 모든 사용자가 이러한 피싱 공격을 주의를 통해서 방어할 수 있다고 단정짓기는 어렵다. 추후 이런 문제도 해결할 수 있는 방안이 요구되며, 전자우편의 특성상 이미지를 포함하는 경우가 많기 때문에 TTP 인증을 통한 워터마킹이나 스테가노그래피를 이용한 전자우편 인증 체계를 포함해야 할 것이다.

참 고 문 헌

[1] APWG, "phishing Activity Trends Report", Jan, 2005.

[2] Miek Gieben, "DNSSEC-Volume 7, Issue 2", THE INTERNET PROTOCOL JOURNAL, June, 2004.

[3] 한국정보보호진흥원(KISA), "IE 등 인터넷 탐색기의 각종 Spoofing 취약점", Nov, 2004.

[4] Roy Arends, Rob Austein, Dan Massey, Matt Larson, Scott Rose, "Resource Records for the DNS Security Extensions", Work In Progress, Oct 10, 2004.

[5] Microsoft, *"http://support.microsoft.com/?id=833786"*

[6] Microsoft, *http://www.microsoft.com/korea/security/incident/spoof.asp"*

[7] Secunia, "Internet Explorer URL Spoofing Vulnerability", *"http://secunia.com/advisories/10395/"*

[8] US-CERT, "Microsoft Internet Explorer DHTML Editing ActiveX control contains a cross-domain vulnerabilit y", *"http://www.kb.cert.org/vuls/id/356600"*

[9] securityfocus, "Microsoft Internet Explorer Modal Dialog Zone Bypass Vulnerability", Jan 11, 2005.

[10] securityfocus, "Microsoft Internet Explorer IFRAME Status Bar URI Obfuscation Weakness", Nov 02, 2004.

[11] securityfocus, "Microsoft Internet Explorer TABLE Status Bar URI Obfuscation Weakness", Nov 02, 2004.

[12] securityfocus, "Microsoft Internet Explorer Spoofed Address Bar Vulnerability", Aug 16, 2004.

[13] securityfocus, "Microsoft Internet Explorer JavaScript Interface Spoofing Vulnerability", Jul 12, 2004.

[14] Mat Bright, "Spoof Email Phishing Scams and Fake Web Pages or Sites", millersmiles, Feb 23, 2004.

[15] Mat Bright, "Spoof Email Hoax scams and Fake Web Pages or Sites", millersmiles, Feb 23, 2004.

[16] NetworkAppliance, inc, "Advisory Note: In Your Company At Risk?", June 6, 2004.

[17] The Honeynet Project & Research Alliance, "Know your Enemy:Phishing", May 16, 2005.

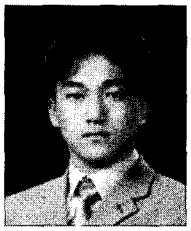
[18] Robert Louis B. Stevenson, "Plugging the phishing hole: legislation versus technology", Computer Crime Research Center, Mar 17, 2005.

[19] Brien Posey, "How to Avoid Phishing Scams", Jan 27, 2005.

[20] J. Zhou and D. Gollmann, "A fair non-repudiation protocol," In Proceedings of 1996 IEEE Sym posium on Security and Pivacy, pages 55-61, Oakland, California, May 1996

[21] K. Kim, S. Park and J. Baek, "Improving Fairness and Privacy of Zhou-Gollmann's Fair Non-repudiation Protocol," IEEE International Workshop on Security, 1999

〈著者紹介〉



민 동 옥 (Dong-og Min) 학생회원
2002년 2월: 고려대학교 전자 및 정보공학과 졸업
2002년 3월~현재: 고려대학교 정보보호대학원 석사과정
〈관심분야〉 네트워크 · 시스템보안, 신경망, 생체인식

손 태 식 (TaeShik Shon) 학생회원
2000년 2월: 아주대학교 정보컴퓨터 공학부 졸업(공학사)
2002년 2월: 아주대학교 정보통신공학과 졸업(공학석사)
2002년 3월~2005년 8월: 고려대학교 정보보호학과 졸업(공학박사)
2005년 8월~현재: 삼성전자 통신연구소
〈관심분야〉 네트워크 · 시스템보안, 인터넷프로토콜 보안

문 종 섭 (JongSub Moon) 정회원
1981년 2월: 서울대학교 계산통계학과 학사
1983년 2월: 서울대학교 계산통계학과 석사
1992년 2월: Illinois Institute of Technology 박사
1993년~현재: 고려대학교 전자 및 정보공학부 교수
고려대학교 정보보호대학원 겸임 교수
〈관심분야〉 IDS, 신경망, 생체인식, 운영체제