웹 사이트 보안수준 확인을 위한 파일럿시스템

김문정*, 김상록**, 조상현***, 이민수**, 이준섭**, 김인호****,김성훈****, 김영갑*
*고려대학교 정보경영정보공학전문대학원
**KAIST 전자전산학과
***엔에이치엔(주) 보안분석팀
****한국정보보호진흥원 기술정책팀
e-mail: *{tops, always}@korea.ac.kr
**{srkim, mslee, jslee}@dependable.kaist.ac.kr
***bungae@nhncorp.com
****{kih, kimsh}@kisa.or.kr

A Pilot System for Website Security-Level Check

Moon Jeong Kim*, Sang-Rok Kim**, Sanghyun Cho***, Min-Soo Lee**,
Jun-Sup Lee**, In Ho Kim****, Sung Hoon Kim****, Young-Gab Kim*
*Graduate School of Information Management and Security, Center for Information Security Technology (CIST), Korea University
**Div. of Computer Science Dept. of EECS,
Korea Advanced Institute of Science and Technology (KAIST)
***NHN Corporation
****Korea Information Security Agency (KISA)

요 약

최근 몇 년 동안 피싱, 파밍, 크라임웨어에 의한 피해 사례 발생이 증가되고 있다. 현재까지의 피싱 관련 솔루션이 대부분 블랙리스트 방식이고 아직까지 피싱 사이트 판단 기준이 없으며 사람들이 이에 대한 인식의 부족으로 인해 이러한 위협을 대처하는데 많은 한계를 가지고 있다. 이에 본 연구에서는 화이트 리스트 기반 웹사이트 보안수준 확인 시스템을 설계하고 이의 파일럿 시스템을 개발하였다. 각 사이트에 대해 피싱 관련 보안수준을 확인하여 신뢰할 수 있는 사이트들을 선별하고 보안수준 정보를 제공함으로써 안전한 인터넷 이용 기반을 제공할 수 있는 방안이 마련될 것으로 기대한다.

1. 서론

Gartner 조사에 따르면 지금까지 5,700만 미국 사용자가 피싱 이메일을 받았고, 200만 사용자가 피해를 당하였으며, 피싱 메시지 중 5%가 성공했다고 한다[1]. 피싱(phising), 파밍(pharming), 크라임웨어(crimeware)에 의한 피해 사례 발생이 증가되고 있는 실정이다. 현재 이에 대한 실제적인 대비책은 매우 미흡하며, 특히 사용자들의 피싱 사이트 판단 기준 및 의식이 부족한 실적이다. 따라서, 피싱 사이트에 대한 대비책 및 각 사이트 별 피싱사이트 인지에 대한 확인 메커니즘이 절실히 필요하다 [2].

최근 은행권을 중심으로 한 금융업체들이 피싱 관련 솔루션 개발 및 도입을 추진 중에 있다. 의심되는 인터넷 사이트의 신뢰성을 검증해 해당 사이트가 안전한지를 사용자에게 알려주어 피싱사이트로의 접속을 방지하는 것이 피싱 방지 솔루션의 기본 기능으로, 현재 이용되고 있는 피싱방지 솔루션들로는 피싱 사고 이후 보고된 피싱사이트를 확인 및 등록하는 블랙 리스트(Black List) 방식이 대부분이다. 본 연구에서는 피싱 사이트, 악성코드 유포 사이트 등으로부터 웹사이트 이용자를 보호하기 위해 국내 웹사이트의 화이트 리스트 데이터베이스(White List DB)를 구축하고, 이를 통해 안전한 인터넷 이용 기반을 제공할 수 있는 방안을 마련하고자 한다. 이를 위해 웹사이트 보안수준 확인 시스템을 설계하고 파일럿 시스템을 개발하고자 한다.

2. 관련 연구

본 절에서는 현재 피싱 방지를 위한 국내외 솔루션들에 대해 간략히 소개한다.

2.1 국내 솔류션

현재 국내에서 개발되고 있는 안티피싱 솔루션들을 표 1에서 보인다. 화이트얼럿 제품을 제외한 대부분의 안티피싱 솔루션들의 경우, 블랙 리스트를 이용하여 보안 수준(위험도)을 결정하고 있는 상황이다. 화이트얼럿 제품의 경우, 제품 개발을 진행 중에 있기 때문에 세부 정보를 확인할 수 없으며, 제품 개발하여 사용까지는 시간이 걸릴 것으로 예상된다. 그 밖에 국내의 대부분 피싱방지 솔루션들은 아직 개발 중이거나 다른 기능과 함께 어플리케이션 형태로 제공되고 있는 추세이다. 외국의 사례와 비교했을

때 피싱방지 기능을 가진 모듈은 툴바 형태의 어플리케이션으로 제공될 것으로 예상된다.

<표 1> 국내 안티피싱 솔루션

제품명	보안수준 결정	서비스 제공 형태
노피싱[3]	블랙 리스트	컴포넌트 (ActiveX)
시큐플랫 VIP[4]	실시간 IP모니터링	어플리케이션 (IP 모니터링시스템)
화이트얼럿	화이트 리스트	현재 개발 중
노턴컨피덴셜[5]	블랙 리스트	어플리케이션
클라이언트키퍼 피싱프로[6]	DB활용+피싱패턴 및 컨텐츠분석기술	어플리케이션
엔프로텍트 피싱헌터 ASP[7]	보호대상 정보패턴 미리 정의	어플리케이션

2.2 국외 솔루션

현재 국외에서 개발되고 있는 안티피싱 솔루션들을 표 2에서 보인다.

<표 2> 국외 안티피싱 솔루션

제품명	보안수준 결정	서비스 제공 형태
PhishTank[8]	블랙 리스트	어플리케이션 (리포트기능)
Virtual Security™ Web Appliance[9]	블랙 리스트 +행동기반필터링 +URL필터링엔진	어플리케이션
FraudAction[SM][10]	FACC[1] 검증	어플리케이션
TraceAssure[11]	화이트 리스트	툴바
VeriSign[12]	블랙 리스트	보안제품의 일부
NXD[13]	블랙 리스트	사이트검색
OpenDNS[14]	블랙 리스트	사이트검색
TrustWatch[15]	블랙 리스트	컴포넌트 (ActiveX, 툴바)
Netcraft[16]	블랙 리스트	컴포넌트 (ActiveX, 툴바)
Microsoft Phishing Filter[17]	블랙 리스트	툴바

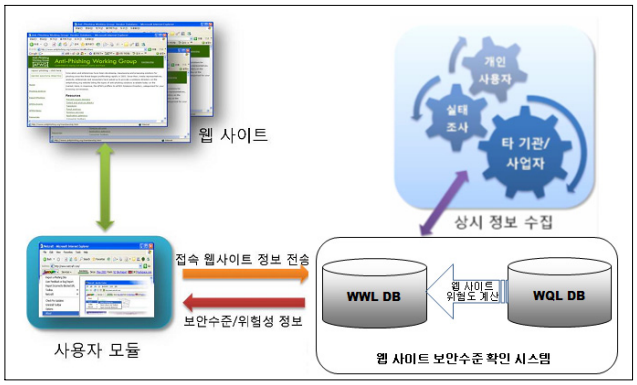
국외의 경우 안티피싱 워킹그룹, APWG (Anti-Phishing Working Group)를 중심으로 피싱 방지에 대한 연구 활동이 활발히 진행되고 있으며, 국외 역시 대부분 블랙리스트를 이용하여 위험 상태(위험도)를 결정한다[18]. 또한 대부분의 솔루션들이 툴바 형태로 제공됨을 알 수 있다.

1) AFCC는 RSA에서 자체적으로 운영하고 있는 Fraud Analysis 팀인 Anti-Fraud Command Center임.

3. 웹사이트 보안수준 확인 시스템

3.1 전체 시스템 구조

본 연구에서 제안하는 웹사이트 보안수준 확인 시스템의 전체적인 구조는 그림 1에서 보이는 바와 같다.



(그림 1) 웹사이트 보안수준 확인 시스템 구조

본 연구에서 제안하는 웹사이트 보안수준 확인 시스템에서는 기본적으로 신뢰 가능한 사이트들의 목록을 관리하는 WWL DB(Website White List DataBase) 와 그것의 후보자 사이트들의 목록을 관리하는 WQL DB(Website Qualified List DataBase)를 사용함으로써 사이트 위험도를 판별한다. 사용자가 페이지를 방문할 때 그 페이지에서 링크가 없는 텍스트 정보만을 보여 주는 경우에는 위험 요소가 없다고 간주되어 바로 출력한다. 그러나 단순 텍스트 페이지가 아닐 경우 우선적으로 WWL DB를 확인하고 WQL DB를 확인하게 된다.

3.2 요구사항

웹사이트 보안수준 확인을 위한 시스템을 구현하기 위해서는 다음과 같은 사항들이 고려되어야 한다.

(1) 클라이언트 측 고려 사항

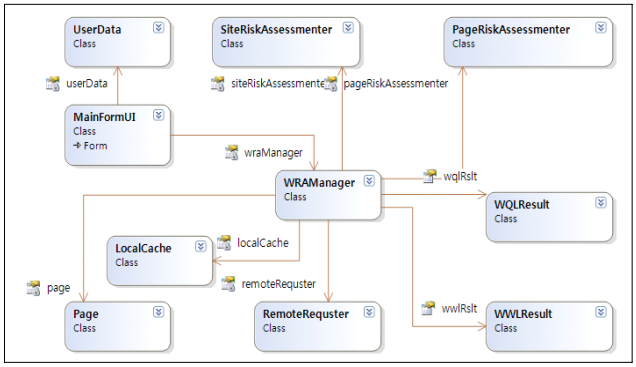
사용자가 요청하는 웹 페이지의 대한 정보의 확인이 가능해야 하며, 사용자에 대한 피드백이 용이해야 한다. 그리고 사용자가 클라이언트 프로그램으로 인해 느려짐으로 인한 불편이 최소화 되어야 한다.

(2) 서버 측 고려 사항

많은 수의 클라이언트의 요청을 효율적으로 처리해야 한다. 그리고 서버 자체에 대한 보안을 강화해야 한다.

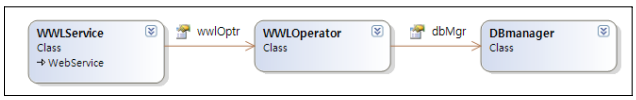
3.3 모듈기반의 시스템 구조

본 연구에서 제안하는 시스템에서는 웹사이트의 위험을 판단하거나 WWL DB를 관리하는 WWL 서버 또는 WQL DB를 관리하는 WQL 서버로 웹사이트의 위험도 측정을 요청하여 사용자에게 피드백을 해 주는 클라이언트가 존재한다. 이러한 기능을 갖는 클라이언트 시스템의 모듈기반 구조는 그림 2에서 보이는 바와 같다.



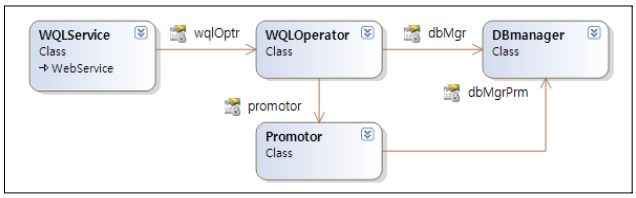
(그림 2) 클라이언트 시스템 구조

웹사이트의 화이트 리스트에 대한 목록을 유지 및 관리하는 WWL 서버의 모듈기반 구조는 그림 3에서 보이는 바와 같다.



(그림 3) WWL 서버 시스템 구조

WWL DB를 구축하기 위해 WQL DB가 사용되며, 이를 유지 및 관리하는 WQL 서버의 모듈기반 구조는 그림 4에서 보이는 바와 같다.

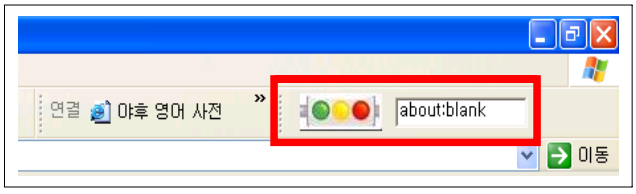


(그림 4) WQL 서버 시스템 구조

4. 파일럿 시스템 개발

4.1 클라이언트 시스템 구현

웹사이트 보안수준 확인을 위한 클라이언트 측 고려사항을 충족시키기 위한 방법으로 본 연구에서는 웹 브라우저의 툴바형태로 웹사이트 보안 수준 확인 시스템의 클라이언트를 제작하였다. 본 연구에서 구현한 클라이언트 모듈은 그림 5에서 보이는 바와 같다.



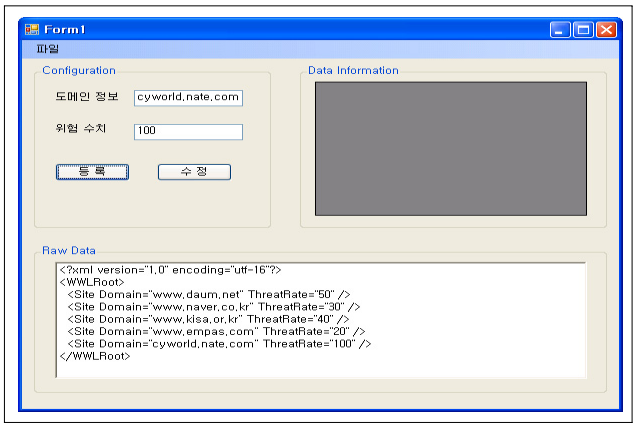
(그림 5) 툴바 형태의 클라이언트 모듈

그림 5에서 보이는 바와 같이, 웹 브라우저의 툴바에서 사용자에게 피드백을 줌으로써, 기존 브라우저의 친숙한 환경을 그대로 활용가능하다. 먼저, 사용자가 요청하는 웹 페이지에 대한 정보의 경우, 웹 브라우저에서 BHO(The Browser Helper Object) 컴포넌트를 활용하여 확인할 수 있다. 사용자에 대한 피드백은 웹 브라우저의 툴바 형태임을 감안할 때 웹 브라우저 자체에 표현할 수 있는 장점을 가진다. 마지막으로 툴바 형태의 클라이언트는 웹 브라우저의 구동 시 모듈형태로 실행되기 때문에 사용자가 거의 느끼지 못할 정도의 자원을 사용한다.

클라이언트 측 모듈로서의 효용성을 가지기 위하여 툴바는 사용자가 이용하는 웹사이트의 정보를 추출할 수 있어야 하며, 사용자가 사용하는 웹 브라우저에 어떠한 데이터라도 표현할 수 있어야 한다. 이러한 기능을 구현하기 위해서 웹 브라우저에서 모듈 추가를 위해 제공하는 인터페이스를 활용하였다. 툴바 작성을 위한 모듈은 MS에서 미리 정의한 몇몇의 클래스(UserControl, IObjectWithSite, IDeskBand, IDockingWindow, IOleWindow, IInputObject)를 상속받아서 구현하였다.

4.2 서버 구현

서버 측의 구현을 위해서는 많은 수의 클라이언트의 요청을 효율적으로 처리하고, 서버 자체에 대한 보안이 강해야 한다는 조건을 만족해야 한다. 이를 위해 WWL DB와 WQL DB를 이용하였다. 또한 실제 많은 웹서버를 사용하고 있는 웹 환경을 고려할 때, 매우 많은 데이터와 클라이언트로부터의 요청을 처리해야 한다. 그림 6에서 서버 측으로 부터 클라이언트 측으로 전달되는 xml의 한 예를 보인다.



(그림 6) 서버 측 데이터 전송 형태

WWL DB 내의 데이터는 그림 6에서 보이는 바와 같이 xml 형태로 클라이언트 측에 전달되며, 이러한 데이터는 사이트의 보안 수준을 확인하는데 이용할 수 있다.

본 연구에서는 Oracle DB 또는 MS SQL DB 와 함께 XML 웹 서비스를 활용하고자 한다. Oracle 및 MS SQL 서버는 많은 사용자를 보유하고 지속적으로 사용되고 있는 만큼 그 안정성 및 보안에 대해 인정을 받았다고 볼 수 있다. 또한 XML 웹 서비스는 기존의 웹 환경을 그대

로 이용하여 데이터를 전송할 수 있는 장점을 가지고 있어, 본 연구에서 제작한 클라이언트 모듈 외의 모듈들과도 호환성을 유지할 수 있다는 장점을 가진다.

5. 결론 및 향후과제

본 논문에서는 최근 이슈가 되고 있는 피싱 및 파밍과 같은 공격들로부터 사용자를 보호하기 위한 솔루션에 대해 분석과 방안을 제시하였다. 기존의 블랙 리스트 기반의 안티피싱 솔루션의 문제를 극복하기 위해 화이트 리스트 기반의 솔루션을 제안하였고, 화이트 리스트를 유지하고 실시간으로 웹사이트의 위험도를 측정하기 위해 웹사이트 보안수준 확인 시스템을 설계 및 구현하였다. 그리고 대표적인 몇몇 웹사이트들을 이용하여 모의실험을 진행함으로서 제안된 시스템과 알고리즘의 유효성을 검증하였다.

제안된 시스템은 피싱 사이트로부터 사용자를 보호한다는 목적을 달성하고 있지만 그 내부의 알고리즘과 실용성 측면을 고려해 보았을 때 아직 고려되어야 할 부분이 존재한다. WWL DB 및 WQL DB를 유지하기 위한 방법 및 알고리즘 연구가 현재 진행되고 있지만 신뢰 가능한 사이트를 분별하기 위한 공신력 있는 기준이 필요하다. 그리고 실제 사용자에게 배포되었을 때 사용자에게 미치는 영향을 최소화하기 위해 클라이언트 모듈의 플랫폼 문제 등도 고려되어야 할 것이다. 따라서 향후에는 이러한 한계를 극복하고 해결해 나갈 수 있는 기술에 대한 연구가 필요하다.

참고문헌

[1] Litan and Avivah, “Phishing Attack Victims Likely Targets for Identity Theft”, in Gartner First Take FT-22-8873, 2004, Gartner Research

[2] R. Dhamija and J. D. Tygar, “The Battle against Phishing: Dynamic Security Skins", Proc. of the 2005 symposium on Usable Privacy and Security, pp. 77-88, Jul. 2005, Pittsburgh, Pennsylvania

[3] nophishing, www.softrun.com/product/nophishing_info.asp, 2007

[4] SecuPlat VIP, http://www.inzen.com/tt/board/ttsite.cgi?act=doc&id=2_09, 2007

[5] Norton Confidential, http://www.symantec.com/ko/kr/home_homeoffice/products/features.jsp?pcid=ts&pvid=nco

[6] PhishingPro, www.softforum.co.kr/kor/down/SoftForum-Brochure (kr)_new.pdf, 2007

[7] PhishingHunter ASP, www.nprotect.com/v6/service/index.php?mode=service _personal#, 2007

[8] phishtank, http://www.phishtank.com/api_documentation.php, 2007

[9] Virtual Security, http://www.finjan.com/objects/whitepapers/Phishing_ WP_Jan07.pdf, 2007

[10] FraudAction, http://www.rsa.com/node.aspx?id=3020

[11] TraceAssure, http://www.tracesecurity.com/products/siteassure_toolbar.php, 2007

[12] VeriSign, http://www.verisign.com/static/005561.pdf

[13] NXD, http://www2.simplicita.com/product_nxd.html

[14] OpenDNS, http://www.opendns.com/features/phishing

[15] TrustWatch, http://toolbar.trustwatch.com/, 2007

[16] Netcraft, http://toolbar.netcraft.com/, 2007

[17] Sharif, T. Phishing Filter in IE7, Sep. 9, 2006. http://blogs.msdn.com/ie/archive/2005/09/09/463204.aspx

[18] Anti-Phishing Working Group. Phishing Archive, http://www.antiphishing.org/phishing_archive/phishing_archive.html