검색 엔진 기반의 안티 피싱 기법

이민수 이형규 윤현수

한국과학기술원

mslee,hklee,yoon@nslab.kaist.ac.kr

An Anti-Phishing Approach based on Search Engine

Minsoo Lee Hyeonggyu Lee Hyunsoo Yoon

Korea Advanced Institute of Science and Technology

요 약

피싱은 인터넷을 이용한 일종의 사기 수법이다. 피싱을 방지하기 위하여 웹 브라우저 밴더에서는 블랙 리스트 기반의 피싱 탐지 기술을 제공하고 있다. 또한 기계학습을 이용한 피싱 탐지 기법들이 제안되어 피싱 공격에 대응을 하고 있다. 하지만, 피싱 공격이 진화 함에 따라 기존의 기술들이 탐지 못하는 경우가 발생을 한다. 피싱 페이지가 생성된 후 일정 시간이 지나지 않을 경우는 기존의 리스트 기반의 솔루션이 탐지를 하지 못하며, 이미지 기반의 피싱 공격의 경우 기존의 연구들이 탐지 하지 못한다. 이에 본 연구에서는 구글 검색엔진을 이용하여 이러한 문제점들을 해결하는 방안을 제안한다.

1. 서 론

네트워크 사용이 증가함에 따라 다양한 형태의 공격들이 사용자들을 위협하고 있다. 이 중에서 특히 피싱 공격은 사용잗르의 민감한 정보를 유출하는 공격으로 매우 위험한 공격 유형으로 분류된다.

피싱 공격으로부터 사용자들을 보호하기 위해 다양한 형태의 연구가 진행되었고, 실 생활에서 사용되고 있는 웹 브라우저에 적용되거나, 독립적인 어플리케이션으로 제공되고 있다.

하지만, 실제 웹 브라우저에 적용되어 있는 방식은 블랙 리스트 기반으로 리스트에 등록되지 않은 공격의 경우는 탐지 할 수 없다[1][2][3]. 또한 학계에서 제안되었던 기계학습을 이용한 피싱 탐지 방식들의 경우 진화된 피싱 공격의 형태인 이미지 기반의 피싱 공격들을 탐지하는데 한계점을 가진다[4][5][6][7].

본 논문에서는 기존의 연구들의 한계점을 극복하기 위해 구글 검색 엔진을 기반으로 피싱을 탐지하는 방안에 대해서 제안한다. 제안된 방식은 내부적으로 리스트를 가지지 않으며, 새로운 형태로 진화 된 이미지 기반이 피싱 공격들 또한 탐지 할 수 있는 방식이다.

이를 위해 2장에서는 관련연구들과 그 한계점을 기술하고, 3장에서는 본 연구에서 제안하는 시스템에 대해 기술하고, 4장에서는 실제 구현방안에 대해서 기술한다. 마지막으로 5장에서는 결론을 기술한다.

2. 관련 연구

피싱을 방지하기 위한 접근 방식은 크게 3가지 접근 방식으로 분류 할 수 있다. 리스트 기반의 접근 방식과 기계학습을 통한 피싱 탐지 기술이다. 또한 리스트 기반의 접근 방식은 저장하는 리스트를 기반으로 화이트 리스트 기반의 접근 방식과 블랙 리스트 기반의 접근 방식으로 분률 할 수 있다.

블랙 리스트 기반의 접근은 현재 대부분의 웹 브라우저에서 제공하는 방식으로 피싱 페이지로 등록되어 있는 URL DB를 바탕으로 사용자에게 피드백을 하는 방식이다. 즉 등록된 피싱 페이지의 리스프를 바탕으로 정보를 제공한다[1][2][3].

화이트 리스트 기반의 접근은 신뢰 할 수 있는 사이트의 도메인 또는 URL을 저장하여, 사용자들이 접근하는 사이트가 안전한 사이트인지 여부를 제공하는 방식이다. 일반적으로 대표적인 사이트들에 대한 정보를 저장한다[8][9].

기계학습을 통한 접근 방식은 사용자가 접근하는 웹 사이트의 페이지의 특성을 기반으로 피싱 여부를 판단하게 된다. 또한 도메인의 다양한 특성 정보들을 통해 대상 사이트가 믿을만 한지 여부를 판단한다[5][6][7].

리스트 기반의 접근 방식은 잘 정리된 리스트의 확보 여부가 성능을 결정하는 중요한 요소가 된다. 하지만, 피싱 페이지가 DB에 등록되지 않았을 경우 탐지하지 못하는 문제점을 가진다. 또한 이미지 또는 플래시 기반의 웹 페이지들이 등장할 경우 기존의 텍스크 기반의 탐지 기법을 이용하였던 기계학습을 통한 접근 방식 또한 무용지물이 될 수 있다.

따라서 리스트를 구축하지 않으면서, 새로운 형태의 이미지 기반의 피싱 페이지를 탐지할 수 있는 방안이 필요하다.

3. 이미지 기반의 피싱 공격

본 절에서는 진화된 형태의 피싱 공격인 이미지 기반의 피싱 페이지에 대해서 기술한다. 기계학습을

이용한 피싱 탐지 기술들은 대부분 피싱 페이지의 컨텐츠(html 코드, 내용)등을 기반으로 피싱 여부를 판단한다. 하지만 그림 1과 같은 피싱 공격이 발생 할 경우 이러한 기술들을 적용할 수 없게된다.
그림 1은 이미지 기반의 피싱 페이지를 보여주고 있는데, 사용자로부터 정보를 입력받는 폼(form) 태그 부분을 제외한 모든 부분이 이미지로만 구성되어 있다.

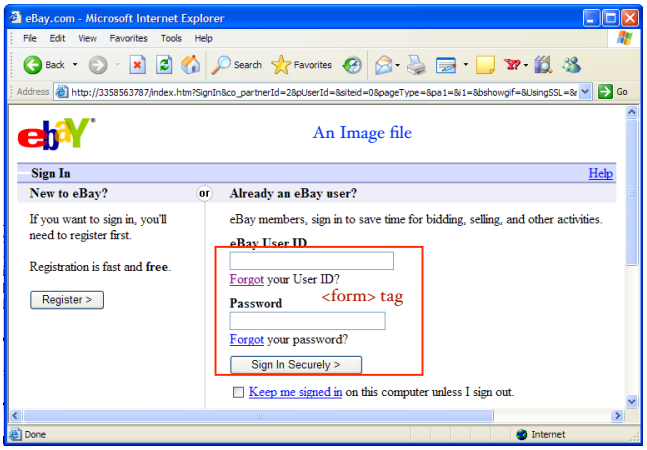

그림 1. 이미지 기반의 피싱 페이지

예컨데, 08년 www에 발표된 CANTINA의 경우 대상 웹 페이지의 중요한 단어를 5개 추출하여 이를 활용하는데, 이 경우 중요한 단어를 찾을 수 있는 방법을 잃게 된다[5]. 기계학습 기반의 연구들 중 피싱 페이지의 컨텐츠에 기반한 탐지 알고리즘은 모두 사용할 수 없게 된다.

3. 검색엔진 기반의 안티 피싱 기법

본 절에서는 기존의 연구에서 가지는 다음과 같은 한계점을 극복하기 위해 검색 엔진 기반의 피싱 기법을 제안한다.

- 피싱 DB를 이용함으로써 발생하는 초기 탐지 문제
- 이미지 기반의 피싱 공격의 한계

3.1 기본 접근 방식

앞서 언급한 두가지 문제를 해결하기 위해서 본 연구에서는 검색엔진의 검색 결과를 활용한다. 검색엔진은 키워드를 입력받아 그 내용과 가장 비슷한 내용을 가지는 웹 사이트의 결과를 보여준다. 이러한 검색엔진의 특성은 블랙 리스트 기반의 접근 방식과 화이트 리스트 기반의 접근 방식 처럼 내부적으로 미리 리스트를 구축해야 하는 문제점을 해결해준다. 즉, 정확한 키워드를 추출하여 검색엔진을 활용한다면, 사용자가 들어가고자 했던 페이지인지 여부를 충분히 확인해 줄 수 있다.
새로운 유형의 공격인 이미지 기반의 피싱 페이지에 대응하기 위해서 본 연구에서는 각 페이지를 대표적으로 표현 해 줄 수 있는 키워드인 <title> 태그 내에 있는 정보를 활용한다. <title> 태그는 html의 컨텐츠(<body>) 부분과는 별로로 사용된다. 즉 컨텐츠 부분이 이미지라 하더라도, 웹 브라우저의 맨 윗 부분을 피싱 대상 페이지와 동일하게 표현해주기 위해서는 공격자는 반드시 <title> 태그를 활용 할 수 밖에 없다. <title>에 해당하는 부분은 그림 1에서 "eBay.com" 에 해당한다. 피싱 페이지는 모방 대상이 되는 웹 사이트와 모양이 동일하게 보여지도록 만드는 특성을 가진다. 이러한 관점에서 <title>태그의 사용은 단순하면서도 효율적인 방안이 될 수 있다.

본 연구에서 제안하는 방식은 그림 2와 같은 전체 구조를 가진다

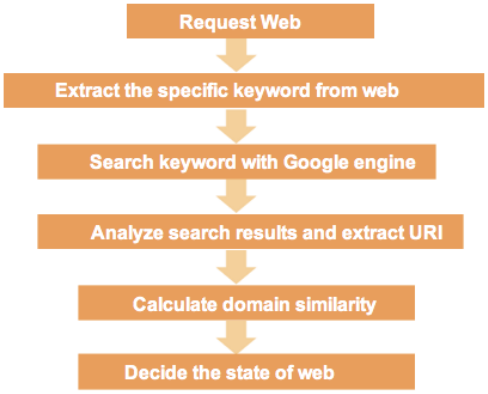

그림 2. 시스템 작동 순서

본 연구에서 제안하는 시스템의 작동 순서는 다음과 같다. 먼저 사용자는 웹 페이지에 접근을 한다. 웹 브라우저에 요청된 웹 페이지가 로드된 후, 본 연구의 시스템은 로드된 페이지가 피싱인지 여부를 판단한다. 앞서 언급한 바와 같이 웹 페이지의 <title> 태그 내에 들어 있는 키워드를 추출하고, 이를 구글 검색엔진을 통해 관련 정보를 검색한다[12]. 특정 임계치의 검색 결과에 포함된 도메인을 분석하여, 현재 사용자가 접속하고 있는 서버와의 연관성을 분석하고, 그 결과를 이용하여 피싱여부를 판단한다.

3.2 검색엔진 결과 비교 알고리즘

본 연구에서는 <title>태그의 내용을 기반으로 검색엔진을 활용하여 그 결과를 현재 접속 중인 웹 페이지의 정보와 비교하게된다. 이는 검색엔진 내에 캐시되어 있는 결과를 비교함으로써, 피싱 페이지 여부를 확인하는 과정이다. 이 과정에서 도메인의 특성을 고려한 비교 알고리즘이 필요한다. 예컨데,

도메인은 2차, 3차, 4차,.., 등 추가적으로 서브 도메인을 가진다. 대부분 피싱의 대상이 되는 유명한 도메인의 경우 2차, 3차 도메인을 사용하여 웹 서버를 관리하고 있다. 이에 검색 결과와 해당 회사의 서버와의 관계성을 적용하기 위하여 본 연구에서는 LCS(Longest Common Subsequence) 알고리즘을 이용하여 URL내의 도메인을 비교한다[11].

LCS 알고리즘을 사용함으로써 2차도메인 3차 도메인 간의 연관관계를 적용할 수 있고, 이는 true positive를 줄이는 효과를 도출할 수 있다.

LCS에 의해 유사도를 비교하는 수식은 다음과 같다.

Similarity Score =
(The number of common subsequence / Length(The shortest string)) X 100

그림 3은 “paypal.com” 과 “login.paypal.com”을 비교하는 LCS 알고리즘을 나타내고 있다.

		l	o	g	i	n	.	p	a	y	p	a	l	.	c	o	m
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
p	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
a	0	0	0	0	0	0	0	1	2	2	2	2	2	2	2	2	2
y	0	0	0	0	0	0	0	1	2	3	3	3	3	3	3	3	3
p	0	0	0	0	0	0	0	1	2	3	4	4	4	4	4	4	4
a	0	0	0	0	0	0	0	1	2	3	4	5	5	5	5	5	5
l	0	1	1	1	1	1	1	1	2	3	4	5	6	6	6	6	6
.	0	1	1	1	1	1	2	2	2	3	4	5	6	7	7	7	7
c	0	1	1	1	1	1	2	2	2	3	4	5	6	7	8	8	8
o	0	1	2	2	2	2	2	2	2	3	4	5	6	7	8	9	9
m	0	1	2	2	2	2	2	2	2	3	4	5	6	7	8	9	10

그림 3. LCS 알고리즘 비교 예

그림 3에서 The number of common subsequence 는 10이되고 paypal.com 의 문자열 길이 또한 10이 된다. 따라서 예제의 경우 100의 similarity score를 가지게 된다.

4. 시스템 구현

4.1 구현 환경

본 연구에서 제안한 시스템은 웹 컨텐츠에 쉽게 접근이 가능하며, 사용자에게 친숙한 형태로 피드백을 할 수 있 는 웹 브라우저 툴바 형태로 구현하였다.

그림 4는 본 연구에서 제안한 시스템인 PhishKiller가 IE 7.0에 설치되어 있는 상태를 보여주고 있다. 그림 4의 오른쪽 상단에 표시되는 것을 확인 할 수 있으며, 세부적인 모양은 그림 5에 나타나 있다.

그림 4. PhishKiller 웹 브라우저 툴 바

그림 5. PhishiKiller 웹 브라우저 툴 바 세부 모형

그림 5의 중간에 보이는 “가” 문자열이 현재 접속하는 사이트가 피싱인지 여부를 알려주는 부분이다. 정상적인 웹 페이지로 판단 될 경우 “가”라는 파란색의 문자열을 보여주게되며, 피싱이라고 판단될 경우 “마”라는 빨간색 문자열이 출력되게 된다. 그외에 다른 부분들은 도메인에 대한 추가적인 특성 정보를 제공해 주는 부분이다. 하지만 본 연구에서 제안하는 방법과는 관련이 없는 부분임으로 설명을 본 논문에서는 관련내용을 기술하지 않는다.

5. 실험 및 검증

본 연구에서 제안한 시스템을 검증하기 위하여 phishtank의 피싱 데이터를 100개를 대상으로 본 연구에서 제시한 알고리즘을 적용하였다[10].

대부분의 피싱 페이지를 정상적으로 탐지 하였다. 하지만, 다음과 같은 경우 오탐이 발생할 가능성이 있음을 확인하였다.

첫번째로, 웹 페이지에 <title>의 정보가 존재 하지 않을 경우 문제가 발생 할 수 있다. 하지만 이 경우 대부분의 유명한 금융 사이트를 분석해 본 결과 모두 <title>태그 내에 내용을 포함하는 것으로 확인되었다. 따라서 개인 사용자의 웹 페이지가 아닌 공식적인 사이트를 모방하는 피싱의 경우 본 연구에서 제안한 시스템의 적용이 가능하다.

두번째로, <title>태그를 사용하지만, 자신 회사에 특화되어 있는 문자열이 아닌 일반적인 문구를 추가 할 경우로 예컨데, “로그인 창”이 <title>에 사용되어 있을

경우 본 연구에서 제안한 시스템을 그대로 적용하게되면 오탐이 발생할 수 있다. 하지만 이러한 경우 비교 대상이 되는 검색 결과의 임계치를 동적으로 증가시킴으로써 이를 보완 할 수 있으며, 키워드를 <title>내의 정보 뿐만 아니라, 현재 접속 중인 도메인 이름과 함께 검색 키워드로 사용하고, 검색 엔진 내의 캐시된 페이지의 수를 활용한다면 이를 보완할 수 있다.

6. 결론 및 향후 연구

본 연구에서는 기존의 피싱 방지를 위해 실제 사용되고 있는 연구들의 문제점들을 분석하고, 이러한 문제점들을 해결 할 수 있는 방안에 대해서 제안하였다. 본 연구에서는 실용적인 측면에서 접근하여 단순하면서도 효과정인 방안인 검색엔진을 활용한 방안에 대해서 제안하였다. 제안된 접근 방식이 검색엔진에 의존적인 부분이 있지만, 검색엔진의 성능이 지속적으로 향상되는 추세를 볼 때, 이는 단점이 아니라 장점이 될 수 있다. 또한 새로운 피싱 공격의 형태인 이미지 기반의 피싱 을 탐지 하기 위하여 컨텐츠 내에서 키워드 추출하는 것 대신에 단순하게 <html>태그의 특성인 <title>태그 내에 컨텐츠를 활용하는 방안에 대해서 제안하였다. 이는 정확도 측면에서 기존의 기계학습을 이용한 탐지 방식보다 부족할 수 있지만, 파라미터 세팅등 추가적인 알고리즘등을 이용하여 이를 극복 할 수 있었다. 또한 기존의 연구에서 탐지하기 어려운 이미지 기반의 피싱을 탐지 할 수 있는 하나의 방안으로써 활용 가능하다는 부분에서 적절한 접근 방식이라고 판단된다. 향후 지속으로 진화되는 피싱 공격들의 특성들을 분석하고 이에 대응할 수 있는 새로운 방안들에 대한 연구가 진행되어야 할 것이다.

참고 문헌

[1] Phishing and Malware Protection, “http://www.mozilla.com/en-US/firefox/phishing-protection/”, visited 2010

[2] Understanding Phishing and Malware Protection in Google Chrome http://blog.chromium.org/2008/11/understanding-phishing-and-malware.html

[3] 신한은행, http://shinhan.com, visited 2010

[4] Anti-Phishing working Group, http://antiphishing.org, visited 2010

[5] Yue Zhang , Jason I. Hong , Lorrie F. Cranor, Cantina: a content-based approach to detecting phishing web sites, Proceedings of the 16th international conference on World Wide Web, May 08-12, 2007, Banff, Alberta, Canada

[6] I. Fette, N. Sadeh, and A. Tomasic. Learning to detect phishing emails. Technical Report CMU-ISRI-06-112, Institute for Software Research, Carnegie Mellon University, June 2006. http://reports-archive.adm.cs.cmu.edu/anon/isri2006/abstracts/06-112.html.

[7] M. Chandrasekaran, K. Karayanan, and S. Upadhyaya. Towards phishing e-mail detection based on their structural properties. In New York State Cyber Security Conference, 2006.

[8] JungMin Kang , DoHoon Lee, Advanced White List Approach for Preventing Access to Phishing Sites, Proceedings of the 2007 International Conference on Convergence Information Technology, p.491-496, November 21-23, 2007

[9] Young-Gab Kim, Sanghyun Cho,Jun-sub Lee, Min soo Lee, In Ho Kim, Sung Hoon Kim, “Method for Evaluating the Security Risk of a Website against Phishing Attacks”, Intelligence and Security Infomatics, 2008

[10] Phishtank, http://phishtank.com, visited 2010

[11] Longest common subsequence Problem,” http://en.wikipedia.org/wiki/Longest_common_subsequence_problem“, visited 2010

[12] Google search engine, http://www.google.com, visited 2010