碩士學位論文

다중링크와 이미지를 이용한 웹사이트 위·변조 탐지기법 연구

A Study on the Website Forgery/Falsification Detecting Technique using Multi-link and Images

2017年 2月

韓南大學校 大學院

컴퓨터工學科

李 한

다중링크와 이미지를 이용한 웹사이트 위·변조 탐지기법 연구

指導敎授　李 克

이 論文을 碩士學位論文으로 提出함

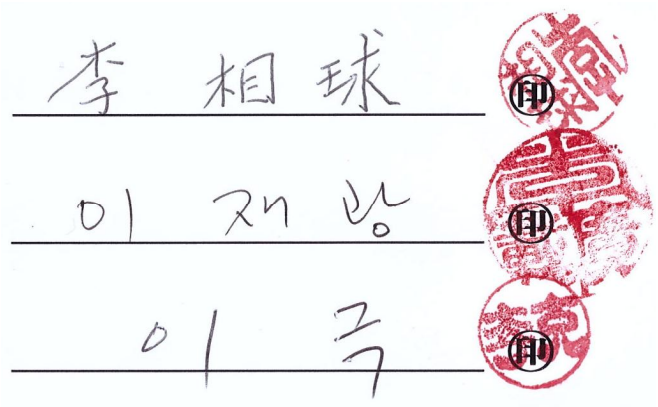
2017年　2月

韓南大學校 大學院

컴퓨터工學科

李　　한

李 한의 碩士 學位論文을 認准함



審査委員長 李 相 球 ㊞

審査委員 이 재 남 ㊞

審査委員 이 근 ㊞

2017年 2月

韓南大學校 大學院

감사의 글

대학원에 발을 들인 후 2년의 시간이 흘러 석사과정을 마치게 되었습니다. 석사과정의 2년을 돌이켜 보면 이것저것 더 배우며 연구하고 싶은 아쉬움이 남습니다. 석사과정동안의 많은 연구가 헛되지 않도록 앞으로의 삶에 좋은 씨앗으로 삼겠습니다.

석사과정의 시작부터 졸업논문의 완성까지 많은 가르침을 주시고 열의와 성의로 지도해 주신 이극 교수님께 진심으로 감사드립니다. 또한, 논문 심사위원으로 아낌없는 충고와 세심한 배려로 신경써주신 이상구 교수님과 이재광 교수님께 깊이 감사드리며, 학부과정 및 석사과정을 마치기까지 신경써주신 소우영 교수님, 박우전 교수님, 이강수 교수님, 최의인 교수님, 이만희 교수님께도 감사드립니다.

예술문화학과 장학조교라는 2년간의 적지 않은 시간동안 많은 편의를 봐주신 예술문화학과 변상형 교수님께도 감사드립니다.

석사과정 동안 많은 부분 신경써주신 육상조 선배님과 김시정 선배님께 감사드리며, 연구실에서 작은 일에도 많은 도움을 주셨던 지호형, 대학원 생활의 절반을 함께한 지용이, 연구실에서 동기로 2년간 함께한 정민이, 연구실은 다르지만 도움과 위안이 된 대학원 동기 준형이, 준현이, 준우, 유진이, 다른 연구실임에도 먼저 다가와 많은 조언과 도움을 주신 래영이형, 재필이형, 재광이형, 이외에도 대학원생활에 큰 도움주신 많은 선배님, 후배님들께 감사드립니다.

마지막으로 석사과정동안 학업에 집중할 수 있도록 지원을 아끼지 않고 응원해주신 존재만으로도 감사한 부모님께 이 논문을 바칩니다.

2017년 2월

이한 올림

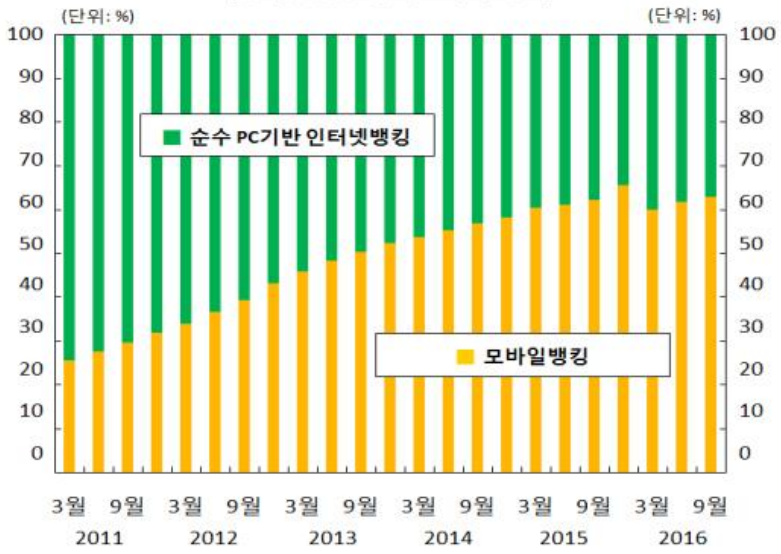
목 차

표 목차

그 림 목 차

제 1 장 서 론

은행에 직접 찾아가 업무를 보던 시대를 지나 집에서 간편하게 은행 업무를 볼 수 있는 시대가 된 후에 온라인을 활용한 은행업무의 편리함으로 인하여 2016년 9월말 인터넷뱅킹서비스와 모바일뱅킹 등록 고객 수는 1억 2,072만 명이 사용하는 거대한 서비스가 되었다. 인터넷뱅킹과 모바일뱅킹의 이용규모(일평균)는 8,790만 건으로 상당히 많은 수가 사용함을 알 수 있다.



[그림 1-1] 인터넷 뱅킹 등록고객 구성비

현재 인터넷 뱅킹보다는 언제 어디서나 장소의 제약을 덜 받으며 예금과 대출금 조회 및 계좌 이체 등 각종 거래를 간편히 할 수 있

는 모바일뱅킹서비스가 더 활발해진 상태이다.

아래의 [표 1-1]과 [표 1-2]는 인터넷뱅킹과 모바일뱅킹의 전체 등록 고객 수와 실제 이용하는 고객 수를 나타내 주고 있다. 이 표에서 알 수 있듯이 모바일뱅킹서비스가 활발해진 지금 현재에도 인터넷 뱅킹의 고객 수와 사용량은 상당히 많은 수임을 알 수 있다.[1]

(천명, 천개, %)

	2015		2016				
				6월		9월	
	9월	12월	3월	전체	실이용 고객수[3]	전체	실이용 고객수[3]
개 인	108,369	109,760	112,485	111,680	51,928	**113,129**	**52,295**
	(1.7)	(1.3)	(2.5)	(-0.7)	(-4.2)	**(1.3)**	**(0.7)**
법 인	6,919	7,093	7,285	7,387	3,322	**7,591**	**3,377**
	(2.9)	(2.5)	(2.7)	(1.4)	(4.5)	**(2.8)**	**(1.7)**
합 계	115,288	116,853	119,771	119,067	55,250	**120,720**	**55,672**
	(1.8)	(1.4)	(2.5)	(-0.6)	(-3.7)	**(1.4)**	**(0.8)**

[표 1-1] 인터넷뱅킹 등록고객 수

(천명, %)

	2015		2016				
				6월		9월	
	9월	12월	3월	전체	실이용 고객수[3]	전체	실이용 고객수[3]
스마트폰 기반	60,075	64,791	68,003	69,768	41,967	**72,030**	**43,774**
	(4.4)	(7.9)	(5.0)	(2.6)	(5.4)	**(3.2)**	**(4.3)**
IC칩·VM 방식	11,801	11,770	3,918	3,841	124	**3,828**	**121**
	(-0.2)	(-0.3)	(-66.7)	(-2.0)	(-2.4)	**(-2.3)**	**(-2.4)**
합 계	71,876	76,561	71,921	73,609	42,092	**75,858**	**43,896**
	(3.6)	(6.5)	(-6.1)	(2.3)	(5.4)	**(3.1)**	**(4.3)**

[표 1-2] 모바일뱅킹 등록고객 수

모바일과 마찬가지로 인터넷뱅킹서비스는 실제이용 고객 수가 많아 해커들의 위협에 쉽게 노출되어 있다. 해커들은 실제 금융 사이트와 유사한 피싱(Phishing)사이트를 이용한 파밍(Pharming)공격으로 금융거래가 가능할 정도의 개인정보를 탈취한다. 일반 사용자들은 피싱사이트가 실제 사이트와 상당히 유사하기 때문에 직접 위·변조 여부를 판단하기가 어렵다. 피싱사이트의 위협에 대응하기 위한 국정원의 국가사이버안전센터, 한국인터넷진흥원의 인터넷침해대응센터 등이 있으나 지속적으로 많은 수의 피싱사이트가 생겨남에 따라 탐지 및 대응 하는데 있어 한계가 있다.

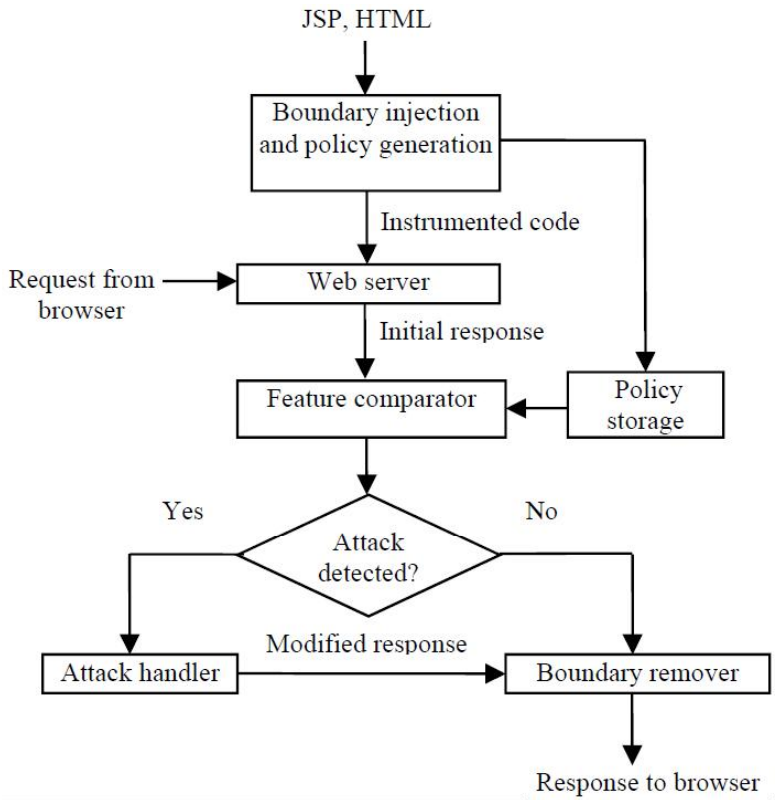
본 논문에서 웹사이트의 위·변조 여부를 탐지하기 위하여 다중링크와 이미지를 이용한 웹사이트 위·변조 탐지 시스템을 제안한다. 제안 시스템은 사용자가 특정 금융 사이트에 접속 시 URL주소를 확인 후 페이지 내의 부분적인 이미지와 웹페이지 링크데이터를 수집하고 기존 정상 웹페이지 정보와 비교를 한 후 페이지 내에 링크되어있는 다수의 웹페이지 역시 부분적인 이미지와 웹페이지 링크데이터를 수집하고 정상적인 웹사이트와 비교하여 정교하게 위·변조된 사이트를 찾아내는 시스템을 제안한다.

제 2 장 관련 연구

2.1 웹사이트 위·변조 공격 방법

2.1.1 XSS 공격

XSS(Cross Site Scripting) 공격은 공격자가 공격대상의 브라우저에 Script를 업로드 하여 사용자 Session 가로채기, 웹사이트 변조, 악의적인 컨텐츠 삽입과 피싱 공격 등을 한다.[11]



[그림 2-1] XSS attack framework

[그림 2-2]는 XSS 공격의 예를 보여주고 있다. 웹페이지에 실행이 가능한 악성코드를 삽입한 후 사용자에게 악성코드가 삽입된 웹페이지를 이용하게 하여 사용자의 context 정보에서 악성코드를 실행 시키는 기법이다.[11][12]

(1)Stored XSS

Stored XSS는 공격자가 특정게시물에 악성 Script를 업로드 한다. 사용자가 게시물을 클릭하는 경우 기존에 공격자가 업로드 한 Script가 전송된다. 브라우저에서 Script가 실행이 된 후 공격자는 사용자의 쿠키와 세션 등 필요한 정보를 획득한다.[11]

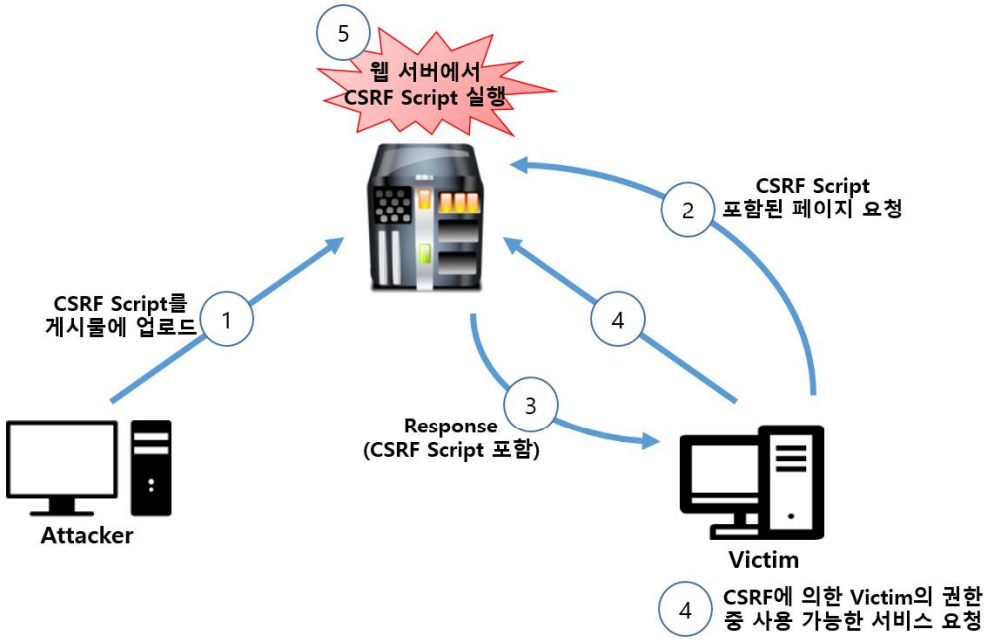
(2)Reflected XSS

URL의 CGI(Common Gateway Interface)의 Parameter에 Script Code를 삽입하는 것으로 이메일을 이용하여 공격자가 사용자에게 특정 웹페이지 링크를 보낸다. 사용자가 링크를 클릭할 때 웹페이지가 화면에 출력 된다. 이때 해당 웹페이지 URL에 삽입된 Script가 실행되며 웹페이지 내용이 변경된다.[11]

2.1.2 CSRF 공격

웹사이트에서 로그인과정을 정상적으로 통과한 사용자를 신뢰한다는 점을 이용하는 CSRF(Cross Site Request Forgery) 공격은 공격자가 사용자인증정보를 악용한다. 변조된 HTTP를 요청하며, 요

청 받은 취약 웹 애플리케이션은 해당요청을 정상으로 인식하여 공격자가 원하는 행위를 수행하는 공격이다.[14]



[그림 2-2] CSRF 공격절차

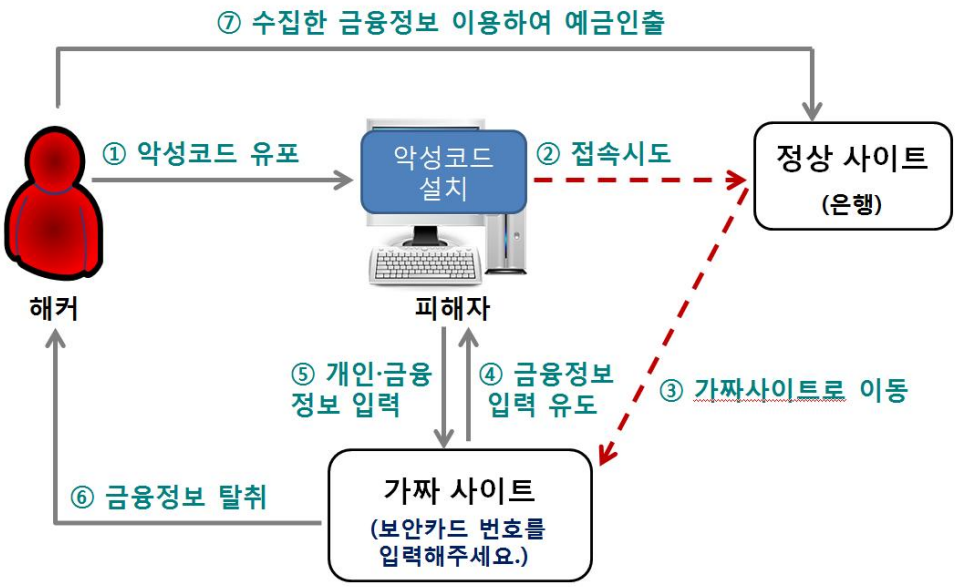
[그림 2-2]는 CSRF 공격절차를 보여주는 그림이다. 첫 번째 과정에서 공격자가 웹 서버에 익스플로잇을 저장하게 된다. 두 번째 과정에서 피해자는 해당 익스플로잇을 포함하는 페이지를 웹 서버에 요청하게 된다. 세 번째 과정에서 웹 서버가 익스플로잇이 포함된 스크립트를 Response 형태로 피해자에게 전송하게 되면 마지막과정에서 피해자의 권한으로 사용가능한 서비스를 웹 서버에 요청한다.[15]

2.1.3 피싱 및 파밍 공격

피싱(Phishing)공격은 개인정보(Private data)와 낚시(Fishing)의 합성어로 공공기관 및 금융기관으로 사칭하여 이메일을 발송하고 가짜은행사이트로의 접속을 유도해 금융정보를 탈취하는 공격이다.

파밍(Pharming)공격은 악성코드를 통해 감염된 사용자 PC를 조작하여 금융정보를 빼내는 방법으로 피해자가 정상적인 사이트에 접속을 시도하더라도 위·변조된 피싱사이트로 접속이 유도되며 각종 금융정보를 탈취하는 공격이다.[10][13][17]

[그림 2-3] 모든 보안카드번호 입력 요구하는 가짜사이트



[그림 2-4] 파밍 공격의 개요

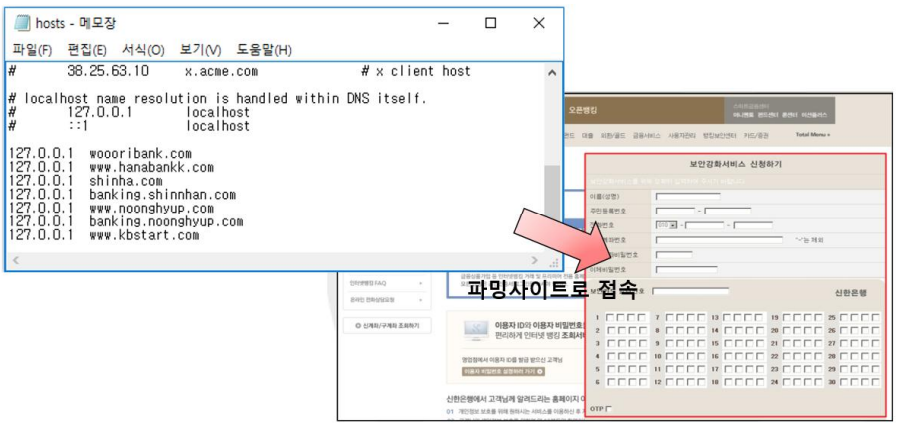
2.1.4 hosts 파일 변조 공격

hosts 파일 변조 공격 방법은 일반적으로 사용자 PC의 호스트 파일을 변조하여 특정 URL에 접속할 때 악성 IP로 연결시키는 공격이다. 아래의 [표 2-1]은 윈도우환경에서 웹브라우저에 URL 입력시 IP Resolving을 수행하는 순서이다.[8]

1	로컬 시스템의 DNS 캐시 정보
2	hosts.ics
3	hosts
4	DAN Query

[표 2-1] IP Resolving 순위

리눅스(Linux)의 경우에는 /etc/hosts 파일이 존재하며, 윈도우즈(Windows)에서는 c:\windows\system32\drivers\etc\hosts 파일이 존재한다. 이 파일들은 IP 주소를 호스트에 매핑 시켜주는 역할을 하는데, 인터넷 통신을 할 때 해당 시스템은 DNS 주소보다 hosts 파일을 먼저 참조하여 원하는 호스트명을 찾은 경우 DNS 서버에 IP 주소를 찾아 달라고 DNS 질의를 요청하지 않게 된다. 공격자는 이점을 이용하여 hosts 파일 내에 공격 대상 URL을 악의적인 IP로 설정하여 사용자가 해당 URL에 접속한 경우 공격자가 설정한 위·변조된 사이트로 접속하게 만든다.[6][8]



[그림 2-5] hosts 파일 변조를 통한 파밍사이트 연결

2.1.5 DBD 공격

DBD(Drive By Download)공격은 공격자가 접속이 활발한 웹페이지를 해킹하여 악성코드를 삽입시킴으로써 변조된 웹페이지에 접속한 사용자는 악성코드를 자신도 모르게 다운로드 받게 되어 감염된

다. 감염된 사용자의 시스템은 사용자가 웹브라우저를 실행 하였을 경우 위조된 웹페이지에 접속하도록 유인하여 금융거래가 가능한 정보를 유출시킨다.[2]

[그림 2-6] DBD 공격에 대한 사용자 감염 경로

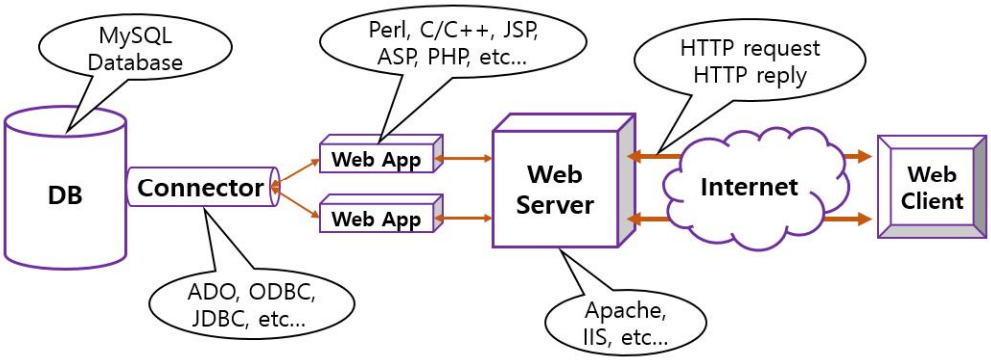
2.1.6 MITM 공격

MITM(Man In The Middle)은 중간자 공격으로 네트워크 통신을 조작하여 통신 내용을 도청하거나 조작하는 공격 기법이다. 통신 중인 두 사람 사이에 중간자가 침입하며, 두 사람은 서로 통신 하는 것으로 생각하지만 실제 두 사람은 중간자에게 연결되어 있어 중간자가 한쪽에서 전송된 정보를 도청 및 조작 후 다른 쪽으로 전송한다.[18]

2.2 위·변조된 웹사이트 탐지 및 대응 기술

2.2.1 웹 방화벽

웹 방화벽(Web Application Firewall)은 일반적인 방화벽과는 달리 웹에 특화된 L7 방화벽이다. SQL Injection, XSS(Cross Site Scripting)등과 같은 웹 서비스 취약점을 이용한 공격을 탐지하고 차단하는 보안 시스템이다. 웹 방화벽은 직접적인 웹 공격대응, 정보유출방지 솔루션, 부정로그인방지 솔루션, 웹사이트 위·변조 방지 솔루션 등으로 활용이 가능하다.[2][19][21]



[그림 2-7] Web Application Architecture

웹에서 이루어지고 있는 대다수의 공격은 웹 애플리케이션(Web Application)을 구축할 때 생긴 취약점을 이용하는데 웹 서버(Web Server)를 공격하거나 DB 내용을 악용하는 방법이 대부분이다. 공격자는 HTTP Request에 특정 공격코드 또는 특정 웹 애플리케이션만이 가지고 있는 취약점을 우회하는 코드를 삽입하여 웹 서버에

전송하게 된다. 결국 웹 애플리케이션은 의도하지 않은 동작을 하게 되고 그 결과를 HTTP Reply를 통해 공격자에게 다양한 정보들을 전송하게 되는 것이다. 웹 방화벽은 웹 서버로 전송되는 HTTP Request Packet을 검사하고 웹 애플리케이션에게 의도치 않은 내용들은 전송되지 않도록 하며, 내용을 감시하여 특정 정보의 유출을 막는 역할도 한다.[20]

2.2.2 URL Spoofing을 통한 피싱 공격 방어 방법

URL(Uniform Resource Location)은 인터넷에서 특정한 정보를 나타내는데 사용되는 주소표현 형식이다. 웹브라우저 취약점, DNS Sniffing등을 이용하는 URL Spoofing은 피싱 공격에 이용가능하다. 공격에 대한 효율적 방안은 사용자가 직접 URL을 확인하는 것이지만, 직접 URL을 확인할만한 판별능력이 많이 부족한 사용자는 피싱 공격에 노출된다. URL Spoofing을 이용하는 피싱 공격 해결을 위한 다른 방법으로는 사용자가 신뢰할 수 있게 위조되지 않은 사이트임을 보이는 것이다. 그래서 특정 사이트를 제3의 기관(Trust Third Party)에서 인증 받아 사용자가 모두 신뢰할 수 있게 하는 것이다.[7][22]

2.2.3 HTTP referer 분석을 통한 피싱사이트 탐지

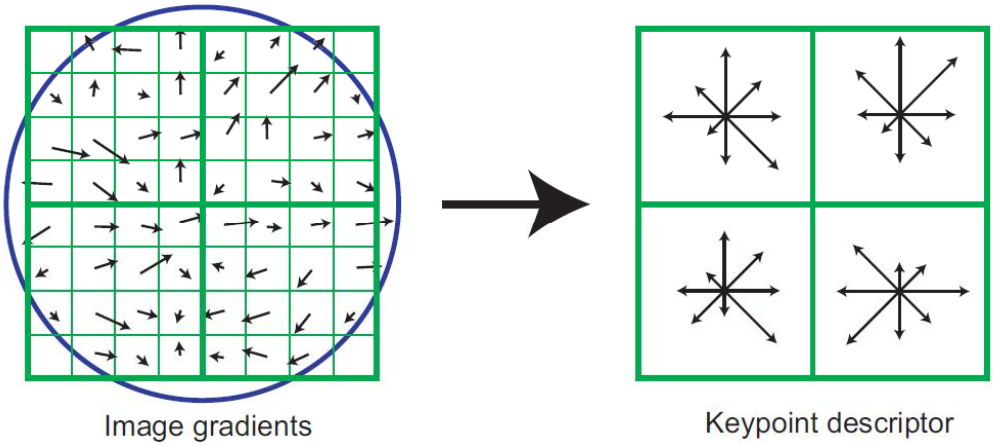
클라이언트에서 서버로 요청을 전송 시 해당 요청은 어느 URL에서 참조되었는지 확인 후 알려주는 기능을 제공하는데, 이것이 HTTP referer header field이다. 즉, 사용자가 웹 브라우저를 이용

해 A 웹사이트를 접속한 상태에서 B 웹사이트로 연결되는 링크를 클릭하여 B 웹사이트로 이동하면 B 웹사이트로 전달되는 HTTP 요청의 헤더부분에는 A 웹사이트의 URL이 입력되어 전달된다. Referer 정보는 현재 웹사이트를 방문한 사용자가 어떤 경로를 이용해 접속했는지 알려주는 역할을 하며 광고, 보안 등 여러 가지 목적으로 사용한다. Referer 정보는 과거 Cross Site Request Forgery 공격 차단을 위한 수단으로도 사용된 바가 있으나 최근에는 referer 조작을 통한 우회기법으로 인하여 사용되지 않고 있으며, 주로 통계 등의 목적을 위해서만 사용되고 있다.[9][23][24]

2.3 이미지 분석 관련자료

2.3.1 SIFT

SIFT(Scale Invariant Feature Transform)는 식별 가능한 키포인트들을 선택한 후에 그 키포인트들을 중심으로 한 로컬 패치(local patch)에 특징 벡터를 추출하는데 모서리나 꼭지점 등에서 벡터로 추출한다. SIFT 특징벡터는 키포인트 주변 픽셀 값을 128차원의 히스토그램으로 나타낸 것으로, image patch를 4x4 블록으로 나눈 후 각각의 블록에 속하는 픽셀의 gradient 방향과 크기에 대하여 히스토그램을 구하고 그 히스토그램 bin 값들을 일렬로 연결하여 나타낸 벡터이다.[25]



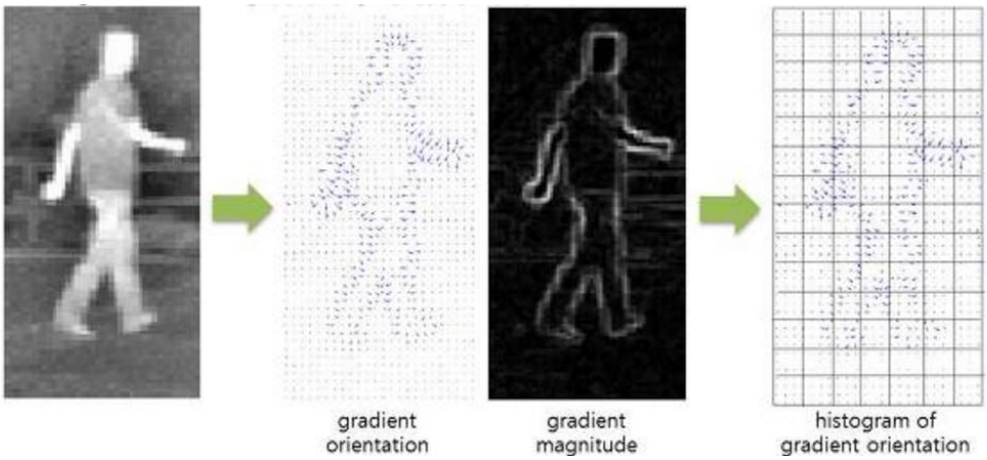
[그림 2-8] 이미지 특징 벡터 추출

[그림 2-8]은 이미지 특징 벡터 추출에 관한 그림으로 왼쪽에서 오른쪽으로 진행 되고, 4x4 블록에서 계산된 2x2 디스크립터 배열을 보여준다. 특징 벡터에서 키포인트 디스크립터는 왼쪽 그림과 같이 키포인트 위치 주변의 각 이미지 샘플 포인트에서 gradient 크기 및 방향을 계산하여 만들어지는데, 이때 중첩 된 원으로 표시되는 가우스 윈도우에 의해서 가중된다. 오른쪽 그림은 왼쪽 그림 4x4 블록에 대한 내용을 요약하여 방향 막대그래프에 2x2 로 나타낸 것으로 화살표의 길이는 영역 내의 해당 방향 근처의 경사 크기 합계에 해당한다.[26][28]

2.3.2 HOG

HOG(Histogram of Oriented Gradient)는 대상 영역을 일정 크기의 셀로 분할한 후, 각 셀마다 edge 픽셀(gradient magnitude가 일

정 값 이상인 픽셀)들의 방향에 대해서 히스토그램을 구하고, 히스토그램 bin 값들을 일렬로 연결한 벡터로, edge의 방향 히스토그램 템플릿으로도 본다.



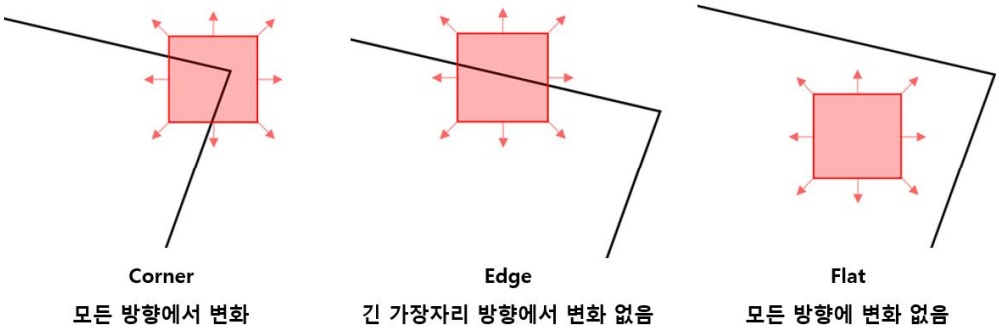
[그림 2-9] HOG 키포인트 획득 과정

HOG는 템플릿 매칭과 히스토그램 매칭의 중간 단계에 있는 매칭 방법으로 볼 수도 있으며 블록 단위로는 기하학적 정보를 유지한다. 블록 내부에서 히스토그램을 사용함으로 로컬한 변화에 일정부분 강인한 특성을 가진다. 그리고 edge의 방향정보를 이용하기에 일종의 edge기반 템플릿 매칭 방법으로 볼 수도 있다. 기본적으로 edge는 영상에 대한 밝기 변화, 조명변화 등에 덜 민감하며, 그에 따라 HOG도 유사한 특성을 갖는다고 할 수 있다. 또한 HOG는 물체의 실루엣(윤곽선) 정보를 이용하므로 사람, 자동차 등 내부의 패턴이 복잡하지 않으면서 고유의 독특한 실루엣 정보를 가지는 물체를 식

별하는데 있어서 적합한 영상 feature이다.[25][27]

2.3.3 Harris Corner

1988년에 C. Harris가 제안한 특징점(Keypoint) 추출 알고리즘으로 Harris Corner 알고리즘은 Moravec corner 검출 알고리즘의 단점을 개선한 알고리즘이다.

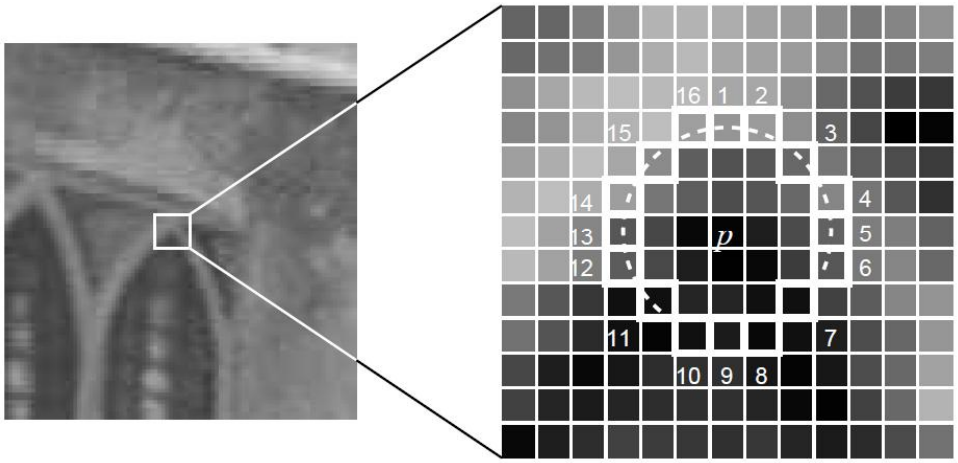


[그림 2-10] Harris Corner Detector

측정방식은 작은 윈도우를 이동시켜서 윈도우 내부 픽셀의 변화되는 값을 이용하여 특징점 추출을 진행한다. [그림 2-10]에서는 Corner, Edge, Flat인 경우를 보여주는 그림으로 상하좌우 이동시 픽셀의 변화되는 값이 커야 Corner 판단을 하게 된다. Flat는 모든 방향에서 픽셀 값의 변화가 없고, Edge에서는 edge가 진행되는 방향으로 이동시에는 픽셀 값의 변화가 거의 없다. Corner는 상하좌우 어떠한 방향으로 이동을 하더라도 픽셀의 변화 값이 큰 것을 볼 수 있다.[32][33]

2.3.4 FAST

Edward Rosten과 Tom Drummond가 제안한 FAST(Features from Accelerated Segment Test) 알고리즘의 기본적인 작동방법은 [그림 2-11]과 같이 임의의 점 p가 Corner인지 여부를 p를 중심으로 원 형태의 16개 픽셀 값을 가지고 판단하며 그중 픽셀의 밝기 값을 이용하고 점 p보다 많이 밝은 경우, 많이 어두운 경우, 유사한 경우의 3가지로 분류한다. 원 형태의 16개 픽셀 중에서 점 p보다 밝은 픽셀이나 어두운 픽셀이 n개 이상 연속되면 점 p를 Corner로 판단한다. 연속되는 n의 값은 픽셀수인 1~16개가 될 수 있으며, FAST 알고리즘은 n의 값을 기준으로 종류가 다양해진다. 여러 종류의 FAST 알고리즘 중 n의 값이 9인 FAST-9가 가장 효율적이고 성능이 좋은 것으로 알려져 있다.



[그림 2-11] FAST Corner Detection

FAST 알고리즘의 문제점으로는 점 p가 Corner로 인식되었다면 점 p와 인접한 점들도 Corner로 검출되는 경우인데, 문제 해결을 위해서 Non-maximal suppression라는 후처리 단계를 적용한다. 식(2-1)을 이용하여 V를 구하고 인접한 Corner 중 자신보다 높은 V 값을 갖는 Corner를 제거한다.[25][35]

$$V = \max\left(\sum_{x \in S_{bright}} |x - p| - t \ , \ \sum_{x \in S_{dark}} |p - x| - t \right) \qquad \text{식(2-1)}$$

2.3.5 BRIEF(Binary Robust Independent Elementary Features)

BRIEF(Binary Robust Independent Elementary Features)는 특징점 Descriptor 생성 알고리즘이다. 여기서 특징점 Descriptor라는 것은 다른 이미지와의 매칭이나 object tracking의 실행을 위한 특징점들 각각에 대한 정보들이다. 주로 FAST 알고리즘을 사용하여 추출한 특징점과 주변의 픽셀들을 가지고 BRIEF 알고리즘을 사용하여 Descriptor를 생성한다. 생성된 특징점 주변 픽셀들을 이용하여 smoothed image patch를 생성하기 위한 과정으로 Gaussian kernel을 사용한다.

$$\tau(p;x,y) := \begin{cases} 1, \text{ if } \quad p(x) < p(y) \\ 0, \ otherwise \end{cases} \qquad \text{식(2-2)}$$

BRIEF 알고리즘에서 이진 서술자를 생성하는데, 식(2-2)를 이용

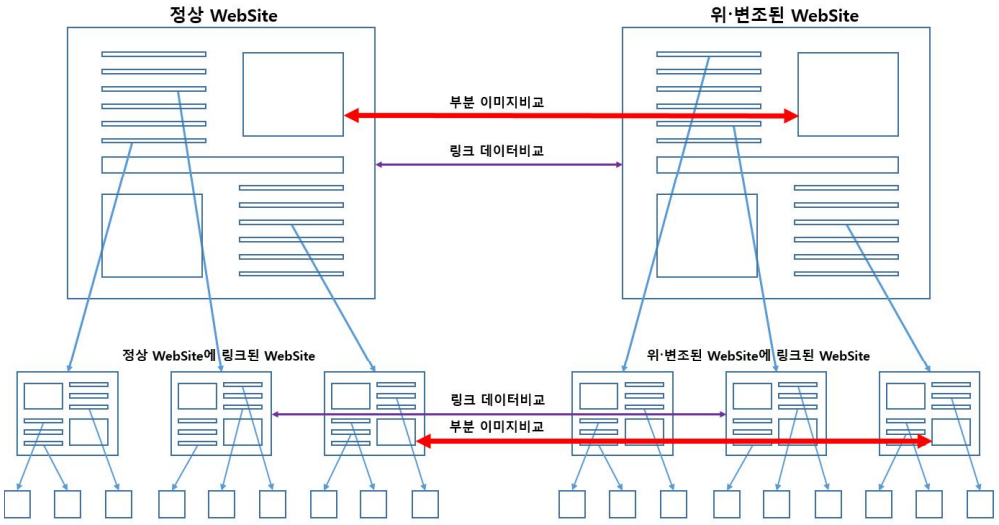
하며 smoothed image patch에서 p의 위치 x, y 한 쌍을 추출하고 픽셀 밝기를 비교하여 x 위치의 픽셀 밝기가 y 위치의 픽셀 밝기보다 작은 경우 1, 그 이외의 경우는 0으로 하여 x와 y에 대한 τ를 판별한다.

$$f_{n_d} := \sum_{1 \le i \le n_d} 2^{i-1} \tau(p; X_i, Y_i) \qquad \text{식(2-3)}$$

x와 y의 판별을 식(2-3)의 내용과 같이 n_d만큼의 반복된 비교 판별을 통한 n_d-bit인 bit string descriptor를 생성하게 된다. BRIEF 알고리즘에서 n_d는 일반적으로 128, 256, 512를 많이 사용한다.[34][36]

제 3 장 위·변조된 웹사이트 탐지 시스템

본 장에서는 다중링크별 이미지를 이용한 웹사이트 위·변조 탐지 시스템을 제안한다. 사용자의 PC를 통하여 주로 이용하는 금융 사이트에 접속 하였을 경우, 제안한 탐지시스템을 이용하여 사용자가 접속한 금융 사이트와 정상적인 웹사이트의 필요한 내용을 수집한다. 수집한 내용을 통하여 비교, 분석한 후 결과 값으로 탐지를 진행하며, 웹사이트의 위·변조 여부를 판별한 후 사용자에게 알려 대처할 수 있게 하고 더 큰 피해로 이어지지 않게 한다.

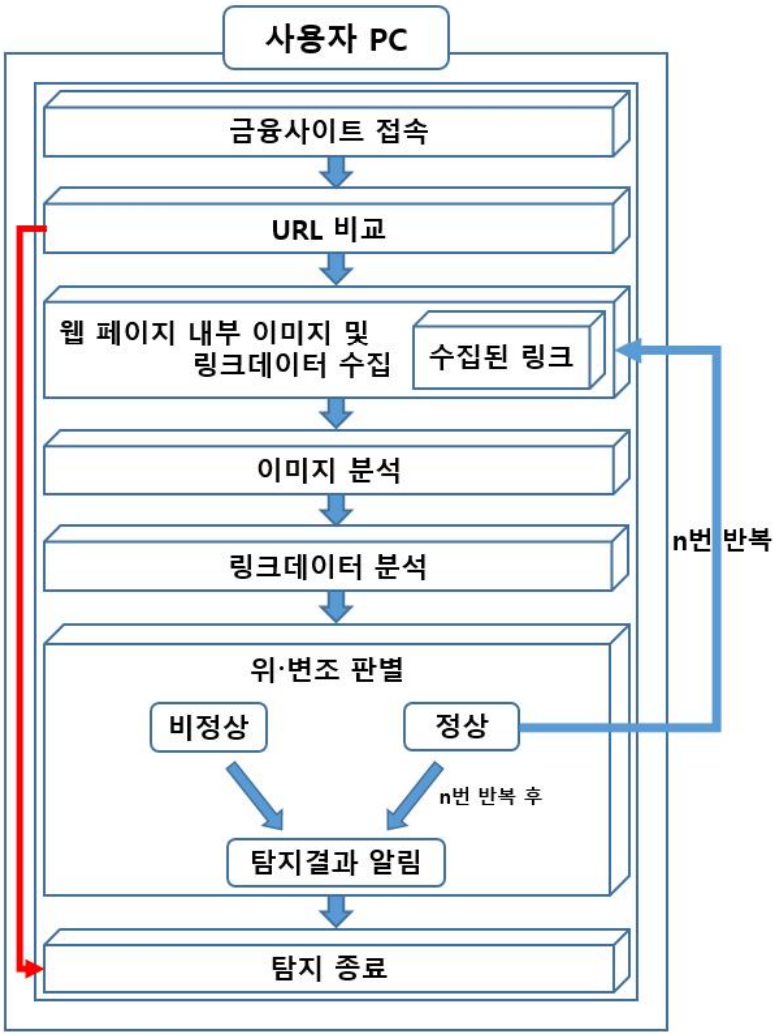


[그림 3-1] 탐지 시스템 트리 구성도

[그림 3-1]은 위·변조된 웹사이트를 탐지하기 위해 본 논문에서 제시하는 시스템의 트리 구성도이다. 현재 웹사이트 최상위 페이지에서 수집한 이미지 및 링크데이터를 정상 웹사이트 최상위 페이지에서 수집한 이미지 및 링크데이터와 비교, 판별하는데, 이 과정을 한 phase라고 정의한다. 한 phase를 수행한 결과가 비정상의 결과 값을 얻게 되면 사용자에게 알리고 탐지 시스템을 종료한다. 결과가 정상의 결과 값을 얻었을 경우에는 링크된 서브페이지로 이동한다. 이동한 서브페이지에서 이미지 및 링크데이터 수집과 비교, 판별을 하는 phase를 수행한다. 이 phase는 사용자가 지정하는 탐지 트리의 n 깊이(depth)까지 반복할 수 있다.

3.1 탐지 시스템 순서도

사용자 PC에서 위·변조된 웹사이트를 탐지하기 위하여 본 논문에서 제안하는 시스템은 사용자가 지정한 n번을 동작한다. [그림 3-2]의 탐지 시스템 순서도와 같이 금융 사이트 접속 후에 탐지 시스템은 5단계(URL 비교, 웹페이지 내부 이미지 및 링크데이터 수집, 이미지 분석, 링크데이터 분석, 위·변조 판별)를 수행하고 반복하며, 수집단계부터 판별단계까지의 단계를 한 phase라 한다.



[그림 3-2] 탐지 시스템 순서도

3.2 탐지 시스템 동작 순서

3.2.1 URL 비교

금융 사이트에 접속한 사용자가 현재 접속한 페이지 URL을 기존에 수집한 정상적인 URL주소들과 비교를 하게 되는데, URL을 비

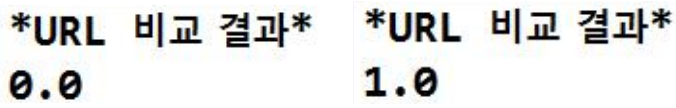
교한 결과 값이 정확하게 일치하는 경우에만 웹페이지 내부 이미지 및 링크데이터 수집 단계로의 진행이 이루어진다. URL을 비교한 결과 값이 정확하게 일치하지 않는 경우에는 진행 중인 탐지시스템을 종료하게 된다. URL 비교단계에서 URL의 문자열을 비교할 때는 Levenshtein Distance 알고리즘을 이용한다.

Levenshtein Distance 알고리즘은 러시아 과학자 Vladimir Levenshtein의 이름에서 유래된 것으로 편집 거리 알고리즘 이라고도 하며, 두 문자열의 유사도를 측정하기 위하여 고안된 알고리즘으로 철자검사, 음성인식, 표절검사 등에 이용한다. 2차원 배열을 이용하여 두 문자열을 비교하며 삽입, 삭제, 변경을 문자열 한부분마다 진행하여 최소편집거리 값을 구한다. 부분마다 구한 최소편집거리의 누적된 값이 두 문자열의 최종편집거리 값이 되며, 그 값을 통하여 유사한 정도를 구하는 판단의 척도로 사용한다.[16]

		A	L	L	I	G	A	T	O	R
	0	1	2	3	4	5	6	7	8	9
E	1	1	2	3	4	5	6	7	8	9
L	2	2	1	2	3	4	5	6	7	8
E	3	3	2	2	3	4	5	6	7	8
V	4	4	3	3	3	4	5	6	7	8
A	5	4	4	4	4	4	4	5	6	7
T	6	5	5	5	5	5	5	4	5	6
O	7	6	6	6	6	6	6	5	4	5
R	8	7	7	7	7	7	7	6	5	4

[표 3-1] Levenshtein Distance 알고리즘 작동과정

[표 3-1]은 ALLIGATOR와 ELEVATOR를 비교하여 편집거리를 구하는 Levenshtein Distance 알고리즘 작동과정으로 같은 값인 경우는 왼쪽 대각선 위쪽의 값을 가져오고 다른 값인 경우는 위쪽, 왼쪽, 왼쪽 대각선 위쪽의 값 중 가장 작은 값에 1을 더한 값을 가져온다. 알고리즘 작동이 끝나면 오른쪽 가장아래 부분의 값이 두 문자열의 편집거리가 된다.



[그림 3-3] Levenshtein Distance 알고리즘을 이용한 URL 비교 결과

[그림 3-3]에서 왼쪽, 오른쪽 그림은 Levenshtein Distance 알고리즘을 이용하여 현재 접속한 페이지 URL 주소와 기존 수집한 URL 주소를 비교한 결과 값이다. 왼쪽 그림은 0.0의 결과 값이 나온 것을 볼 수 있는데, 이 값은 비교하는 두 문자열이 한부분이라도 다른 경우의 결과 값이며, 오른쪽 그림은 문자열이 정확하게 일치하는 경우로 1.0의 결과 값이 나온 것을 볼 수 있다.

3.2.2 웹페이지 내부 이미지 및 링크데이터 수집

웹페이지 내부 이미지 및 링크데이터 수집 단계에서는 이미지 수집과 링크데이터 수집이 진행된다. 이미지 수집 부분에서는 접속한

페이지의 내부에 존재하는 이미지들을 캡처한다. 이렇게 수집된 이미지들은 이미지 분석 단계에 이용된다. 링크데이터 수집부분에서는 HTML 작업을 위한 Java 라이브러리인 jsoup을 이용하여 현재 사용자가 접속한 페이지를 크롤링하여 웹페이지 내에 연결 되어있는 각종 링크데이터들을 수집한다.

```
*링크데이터 수집*
https://www.wooribank.com/#introHome
https://www.wooribank.com/#introNav
https://w1spot.wooribank.com/pot/Dream?withyou=CQIBG0050
https://spib.wooribank.com/pib/Dream?withyou=CMLGN0001
https://sbiz.wooribank.com/biz/Dream?withyou=CMLGN0002
https://www.wooribank.com/#none
https://www.wooribank.com/#content
https://www.wooribank.com/#introMall
https://www.wooribank.com/#introFinance
https://spot.wooribank.com/pot/Dream?withyou=PODEP0001
https://spot.wooribank.com/pot/Dream?withyou=ln
https://spot.wooribank.com/pot/Dream?withyou=fx
https://spot.wooribank.com/pot/Dream?withyou=fn
https://spot.wooribank.com/pot/Dream?withyou=is
https://svc.wooribank.com/svc/Dream?withyou=rp
https://www.wooribank.com/#none
https://spib.wooribank.com/pib/Dream?withyou=CMLGN0001
https://sbiz.wooribank.com/biz/Dream?withyou=CMLGN0002
https://www.wooribank.com/#none
https://spib.wooribank.com/pib/Dream?withyou=ct&fromSite=pib
https://sbiz.wooribank.com/biz/Dream?withyou=ct&fromSite=biz
https://spib.wooribank.com/pib/Dream?withyou=ps
https://spib.wooribank.com/pib/Dream?withyou=PSINQ0001
https://spib.wooribank.com/pib/Dream?withyou=PSTRS0001
https://svc.wooribank.com/svc/Dream?withyou=PSTAX0001
https://spib.wooribank.com/pib/Dream?withyou=PSDEP0010
https://spib.wooribank.com/pib/Dream?withyou=PSFND0001
https://spot.wooribank.com/pot/Dream?withyou=is
https://spib.wooribank.com/pib/Dream?withyou=PSLON0001
https://spib.wooribank.com/pib/Dream?withyou=PSFXD0002
https://spib.wooribank.com/pib/Dream?withyou=PSTRT0086
```

[그림 3-4] 링크데이터 수집 결과

[그림 3-4]는 특정 금융 사이트의 메인페이지를 Java 라이브러리인 jsoup을 이용하여 크롤링한 후의 링크데이터 수집 결과 화면이다. 실험에 이용된 금융 사이트에서 수집된 링크의 수는 250개 이상이었으며 [그림 3-4]의 결과는 그중 일부분이다.

3.2.3 이미지 분석

이미지 분석 단계에서는 이미지 인식과 이미지 검색 등에 사용되며 다양한 이미지 프로세싱 알고리즘을 지원하는 OpenCV를 이용하여 이미지 수집 단계에서 가져온 이미지들을 분석하는데, 그중 ORB 알고리즘을 이용한다.

[그림 3-5] ORB 알고리즘

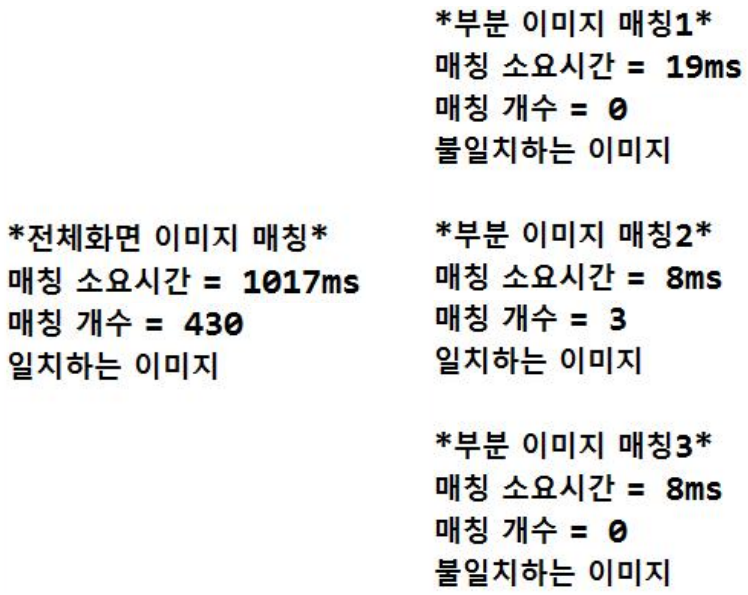
ORB(Oriented Fast and Rotated BRIEF) 알고리즘은 OpenCV(Open Computer Vision) Labs에서 개발된 것으로, FAST(Features from Accelerated Segment Test)[35] 알고리즘과 BRIEF(Binary Robust In dependent Elementary Features)[36] 알고리즘을 기반으로 제안 되었으며, [그림 3-5]는 ORB 알고리즘의 진행순서를 나타낸 것으로 이진 문자열을 이용하여 특징점(Keypoint)을 기술하는 방법이다.[30][31]

Feature matching하여 이미지 일치 여부에 대한 판단을 위해 분석을 진행할 때, 비교분석이 필요한 이미지를 불러온 후 각각의 이미지에 적용할 키포인트를 선언한다. ORB 알고리즘은 키포인트와 디스크립터를 생성 후 이미지의 키포인트 일치를 확인한다. 이때 매칭 소요시간과 매칭 개수(feature), 이미지의 일치 여부를 결과 값으로 얻게 된다.

[그림 3-6] 우리은행 웹사이트 메인 이미지

[그림 3-7] 우리은행 웹사이트 부분 이미지

[그림 3-6]은 특정 금융 사이트의 메인페이지 전체 화면을 캡처한 것으로 왼쪽 이미지와 오른쪽 이미지에 붉은 네모로 표시된 부분을 확인하면 서로 다름을 확인할 수 있다. [그림 3-7]은 메인페이지의 부분 이미지로 기존 이미지 수집단계에서 캡처한 것을 가져온 것이다.



[그림 3-8] 전체 및 부분 이미지 분석

[그림 3-8]은 전체 및 부분 이미지 분석을 한 결과로 [그림 3-6]이 전체화면 이미지 매칭 부분에 해당하며 매칭 소요시간은 1017ms, 매칭 개수는 430으로 일치하는 이미지라고 판단한 것을 볼 수 있다. 그러나 현재 비교중인 이미지는 서로 다른 부분이 존재하는 것으로 확인된 이미지다. 불일치하는 이미지임에도 Feature matching 후 결과 값은 이미지일치로 정확하지 않은 판단을 하는 것을 볼 수 있다. 그래서 전체화면 이미지 매칭 부분에 매칭 개수가 많다고 하여 일치한다고 볼 수 없다는 점을 확인할 수 있다.

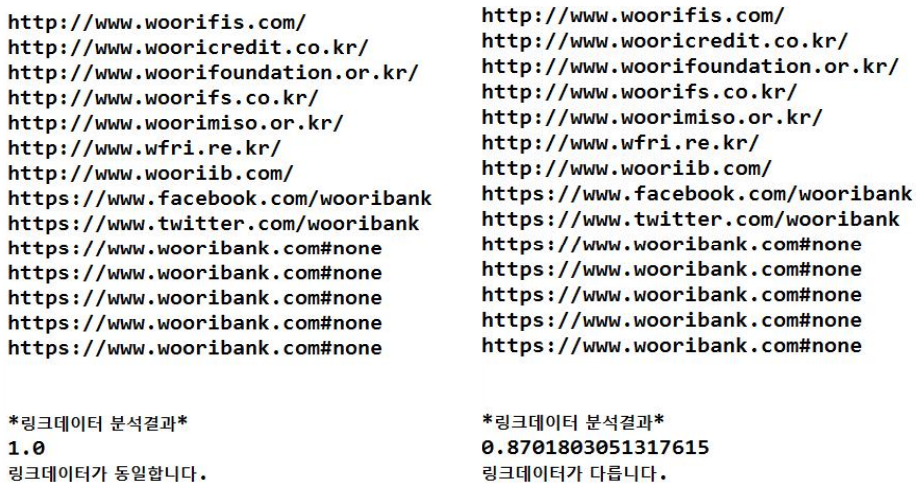
부분 이미지 매칭1은 [그림 3-7]을 분석한 것으로 매칭 소요시간은 19ms, 매칭 개수는 0으로 불일치함을 보였다.

부분 이미지 매칭2, 매칭3은 기존 수집해놓은 다른 위치의 부분

이미지를 비교한 것으로 각각 일치와 불일치를 보였다. 전체화면 이미지 매칭을 통하여 구별할 수 없었던 일치하지 않는 점을 부분 이미지 매칭을 통하여 정교하게 분석을 할 수 있는 점을 확인했으며, 특정 부분의 이미지들만 가져와서 분석함으로 매칭 소요시간도 줄일 수 있었다.

3.2.4 링크데이터 분석

링크데이터 분석단계에서는 Levenshtein Distance 알고리즘을 사용하며 링크데이터 수집 단계에서 수집된 자료들을 정상적인 웹페이지 링크데이터와 비교를 한다. 결과 값으로는 0.0~1.0까지의 값을 얻을 수 있는데, 링크데이터의 비교분석결과가 정확하게 일치할 경우에는 1.0의 값을 갖게 되지만, 일치하지 않을 경우에는 1.0을 제외한 나머지 값이 나타난다.

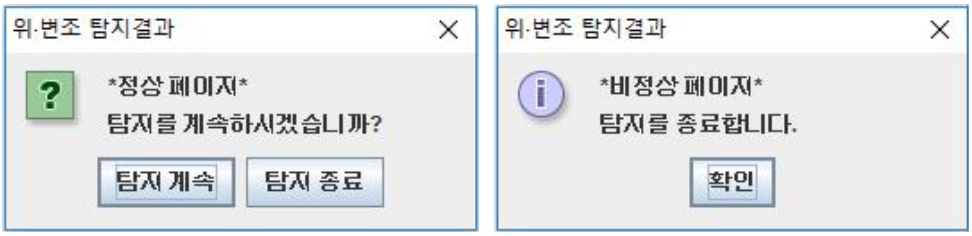


[그림 3-9] 링크데이터 분석결과 화면

[그림 3-9]는 링크데이터 수집단계에서 수집해 놓은 링크데이터들을 가져와 비교 분석한 결과화면이다. 분석 후 결과 값은 1.0에 근접할수록 유사하며, 1.0일 경우에는 동일하다고 판단하는데, 왼쪽 그림의 결과 값은 1.0으로 링크데이터가 동일함을 알 수 있다. 오른쪽 그림의 분석 결과 값은 0.8701803051317615로 정확히 일치하는 값이 아니기 때문에 링크데이터가 다르다는 판단을 하며, 유사한 링크데이터임을 확인 할 수 있다.

3.2.5 위·변조 판별

위·변조 판별 단계에서는 이미지 분석 단계와 링크데이터 분석 단계에서 나온 결과 값을 가지고 웹페이지가 위·변조된 페이지인지 정상인 페이지인지 판단을 진행하게 된다. 판별 단계에서 나온 결과 값이 정상의 범주를 벗어난 비정상 값이면 위·변조된 페이지로 분류하게 되고 탐지를 종료하게 된다. 정상의 결과 값을 얻었을 경우에는 탐지를 종료하지 않고 웹페이지 내부 이미지 및 링크데이터 수집 단계에서 크롤링을 통하여 수집되었던 링크데이터인 서브페이지로 이동하여 내부 이미지 및 링크데이터 수집부터 위·변조 판별 단계까지의 phase과정을 n번을 반복하게 되는데, 탐지시스템의 phase작동 횟수가 증가할수록 더욱더 정교하고 정확한 탐지결과를 얻게 된다.



[그림 3-10] 위·변조 탐지결과 알림 화면

[그림 3-10]은 위·변조 탐지결과를 보여주는 화면으로 왼쪽화면은 정상페이지라고 판단하였을 경우로, 크롤링을 통하여 수집되었던 링크데이터로 이동하여 다시 탐지를 진행할 것인지 탐지를 종료할 것인지 결정하게 된다. 오른쪽화면은 비정상페이지라는 판단이 내려진 화면으로 탐지를 종료하게 된다.

제 4 장 결 론

금융 사이트를 이용한 피싱 및 파밍 공격이 증가함에 따라 일반 사용자가 위·변조를 쉽게 파악할 수 없다는 것이 현실이다.

본 논문에서 웹사이트의 위·변조 여부를 탐지하기 위하여 다중링크와 이미지를 이용하여 위·변조를 탐지하는 시스템을 제안하였다. 제안 시스템에서 URL 비교단계는 사용자가 접속한 현재 금융페이지 URL주소를 기존 수집한 URL주소들과 비교를 진행하였고, 비교할 때는 Levenshtein Distance 알고리즘을 이용하였다. 웹페이지 내부 이미지 및 링크데이터 수집 단계에서는 내부에 존재하는 이미지들을 수집하였으며, jsoup을 이용하여 웹페이지 내에 연결된 각종 링크데이터의 수집을 진행 하였다. 이미지 분석단계에서는 OpenCV Lab에서 개발된 ORB알고리즘을 이용하여 Feature matching을 진행했으며, 이미지의 일치여부를 확인했다. 링크데이터 분석단계에서는 링크데이터 수집 단계에서 수집된 링크데이터를 정상인 웹페이지 링크데이터와 비교, 분석을 진행하였는데, 이 진행과정에서도 Levenshtein Distance 알고리즘을 사용하여 링크데이터의 일치여부를 확인할 수 있었다. 위·변조 판별 단계에서는 이미지 분석과 링크데이터 분석단계에서 나온 결과 값을 종합하여 정상과 비정상을 판단하였다. 비정상의 경우 탐지시스템이 종료되었으며, 정상일 경우 기존에 수집해놓은 링크데이터인 서브페이지로 이동하여 탐지를 진행하였다.

제안한 탐지 시스템의 실험은 2번의 phase에 걸쳐 비교, 수집 탐지를 진행하였다. 제안한 탐지 시스템의 한 phase를 사용자의 조건에 따라 n번 증가 시키면 더욱더 정교한 위·변조 탐지가 가능하다.

향후 연구과제로 제시하는 것은 본 논문의 링크데이터 분석단계에서 확인된 것으로, 기존 정상적인 웹사이트에서 수집한 링크데이터와 여러 환경이 변화한 정상적인 웹사이트의 변경된 링크데이터 값을 비교할 때, 두 비교대상은 정상적인 링크데이터 값을 가지고 있음에도 1.0의 값이 나오지 않고 1.0에 근접한 값이 나오게 되는데, 정상적인 링크데이터 값으로도 다르다는 결과 값이 나오는 경우를 대비하는 연구가 필요할 것이다.

참 고 문 헌

[1] 한국은행, “2016년 3/4분기 국내 인터넷뱅킹서비스 이용현황”, 공보 2016-11-18호, 2016.11

[2] 신지용 “하이퍼링크와 이미지를 이용한 웹사이트 위·변조 탐지 기법 연구” 2016.02

[3] 국가정보원 “http://www.nis.go.kr/”

[4] 김규일, 최상수, 박학수, 고상준, 송중석, ‘이미지 및 코드 분석을 활용한 보안관제 지향적 웹사이트 위·변조 탐지 시스템’, 정보보호학회논문지, 제24권 제5호, pp.871-883, 2014.10

[5] 신지용, 조지호, 이한, 김정민 이극, ‘이미지를 이용한 웹사이트 위·변조 탐지 기법 연구’, 융합보안논문지, 제16권 제1호, pp.81-87, 2016.02

[6] 김봉준 “침입차단시스템에서 DNS 프락시를 사용한 파밍 탐지 기법” 2012.12

[7] 김소영, 강지윤, 김윤정 ‘일반 사용자를 위한 포털 사이트 경유 피싱/파밍 방지 방안’, 한국통신학회논문지, Vol.40 No.06, 2015.06, 1107-1113

[8] 박정민 “사용자를 위협하는 파밍피싱 탐지 기법 제안” 2016.08

[9] 사준호 “HTTP referer 기반 실시간 피싱사이트 탐지 기법”

2012.12

[10] 정연기 “범죄용 인터넷 도메인 특성기반 사이버 범죄 차단 방안” 2014.12

[11] 홍성혁 ‘XSS 공격과 대응방안’, 한국디지털정책학회, 제11권 제12호 pp.327-332, 2013.12

[12] Shahriar, H.; Zulkernine, M. "S2XS2: A Server Side Approach to Automatically Detect XSS Attacks", Dependable, Autonomic and Secure Computing(DASC), 2011 IEEE Ninth International Conference on, vol., no., pp.7,14, Dec. 2011.

[13] 사이버경찰청-정보마당-신용금융범죄, http://www.police.go.kr/portal/main/contents.do?menuNo=200289

[14] 최재영, 이혁준, 민병준, ‘웹사이트 구조와 사용패턴 분석을 통한 CSRF 공격 탐지’, 융합보안논문지, 제11권 제6호, pp.9-15, 2011.12

[15] 이대섭 “웹 애플리케이션에서 전자서명 토큰을 이용한 CSRF 공격 방어 기법 설계 및 구현” 2010.12

[16] Wikipedia , “Levenshtein distance”, http://en.wikipedia.org/wiki/Levenshtein_distance

[17] 사이버경찰청-알림마당-보도자료, “파밍(Pharming) 등 신종금융사기 주의!”, http://www.police.go.kr/portal/bbs/view.do?nttId=14748&bbsId=B0

000011&menuNo=200067&delCode=0 2013.06

[18] 위키백과, “중간자공격” https://ko.wikipedia.org/wiki/%EC%A4%91%EA%B0%84%EC%9E%90_%EA%B3%B5%EA%B2%A9

[19] PentaSECURITY “웹방화벽이란?” https://www.pentasecurity.com/wp/?page_id=305&pn=2&sn=3&sn2=4

[20] 강배근, “웹 방화벽 시스템의 품질 분석에 관한 연구” 2010.06

[21] 정보보안 솔루션, “WAF(Web Application Firewall)″, http://s2kiess.blog.me/220679457699

[22] 민동옥, 손태식, 문종섭, “URL 스푸핑을 이용한 피싱 공격의 방어에 관한 연구”, 정보보호학회논문지, Vol.15 No.5, 2005.10, 35-45

[23] Wikipedia, “HTTP referer”, http://en.wikipedia.org/wiki/HTTP_referer

[24] R. Fielding, UC Irvine, J. Gettys, Compaq/W3C, J. Mogul, Compaq, H. Frystyk, W3C/MIT, L. Masinter, Xerox, P. Leach, Microsoft, T. Berners-Lee, “Hypertext Transfer Protocol – HTTP/1.1”, http://www.w3.org/Protocols/rfc2616/rfc2616.html, Network Working Group, June.1999

[25] http://darkpgmr.tistory.com/

[26] Lowe, D. G., “Distinctive Image Features from Scale-Invariant Keypoints”, IJCV 2004.

[27] N. Dalal and B. Triggs, ″Histograms of oriented gradients for human detection″, CVPR 2005.

[28] Wikipedia, “Gaussian window”, https://en.wikipedia.org/wiki/Window_function#Gaussian_window

[29] 보안뉴스, “지능형 CCTV 구현 기본기술 : 영상기반 객체검출”, 2015.05

[30] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski, “ORB: an efficient alternative to SIFT or SURF”, IEEE International Conference on Computer Vision, pp2564-2571, Nov. 2011.

[31] Lei Yu, Zhixn Yu, and Yan Gong, “An Improved ORB Algorithm of Extracting and Matching Features”, International Journal of Signal Processing, Image Processing and Pattern Recognition, Vol.8 No.5, pp.117-126, May. 2015.

[32] C. Harris and M. Stephens, ″A combined corner and edge detector″, Alvey Vision Conference, 1988

[33] Matching with Invariant Features, Lecture Notes 2004

[34] 서창수, “FAST와 BRIEF 알고리즘 기반의 특징점 추출 하드웨어”, 2015.12

[35] Edward Rosten and Tom Drummond, "Machine learning for high-speed corner detection", ECCV 2006

[36] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua, “BRIEF: Binary Robust Independent Elementary Features”, ECCV 2010

요 약

금융 사이트를 이용한 피싱 및 파밍에 대한 공격이 증가함에 따라 사용자들의 개인정보는 큰 위협을 받게 되었다. 사용자들은 정교하게 위·변조된 웹사이트를 쉽게 알아채지 못한다.

본 논문에서는 웹사이트의 위·변조 여부를 탐지하기 위하여 다중 링크와 이미지를 이용한 웹사이트 위·변조 탐지시스템을 제안한다. 제안 시스템은 사용자가 금융 사이트에 접속 시 URL 주소를 확인, 페이지 내 부분 이미지와 웹페이지 링크데이터를 수집, 기존 정상 웹페이지 정보와 비교하여 정상과 비정상을 판단한다. 정상의 경우 페이지 내에 링크되어있는 다수의 웹페이지도 부분 이미지와 웹페이지 링크데이터를 수집하여 정상적인 웹사이트와 비교한다.

탐지시스템의 횟수를 늘려감에 따라 위·변조 탐지시스템이 정교해지는 점을 보인다.

Abstract

As the number of attacks on phishing and pharming using financial sites has increased, users' personal information has become a big threat. Users can not easily identify Forgery/Falsified Web sites elaborately. In this paper, we propose a Web site Forgery/Falsification detection system using multiple links and images in order to detect whether or not the website is tampered with. When the user accesses the financial site, the proposed system checks the URL address, collects partial image and web page link data in the page, and compares it with the normal web page information to determine normal and abnormal. In the normal case, many web pages linked in the page also collect partial image and web page link data and compare with the normal web site. As the number of detection systems increases, the Forgery/Falsification detection system becomes more sophisticated.