미니페이지 기반 클릭재킹 방지 브라우저의 설계

민재원*, 정성민*, 정태명**
*성균관대학교 전자전기컴퓨터공학과
**성균관대학교 정보통신대학
{jwmin,smjung}@imtl.skku.ac.kr*, tmchung@ece.skku.ac.kr**

Design of the Safe Web Browser Mitigating Clickjacking Attacks Based on Mini-Page

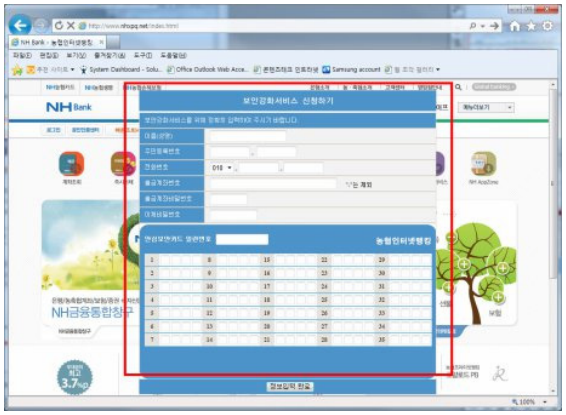
Jae-Won Min*, Sung-Min Jung*, Tai-Myoung Chung**
*Dept of Electrical and Computer Engineering, Sungkyunkwan University
**College of Information Communication Engineering, Sungkyunkwan University

요 약

인터넷 사용이 증가하고 기존의 오프라인 서비스들이 온라인으로 플랫폼을 이동하면서 개인정보나 결제정보 등을 탈취하기 위한 악성 사이트들이 등장하기 시작했다. 공격자들은 피싱 사이트를 만들어 사용자들의 정보 입력을 유도하고 이를 이용해 경제적 이익을 취득한다. 클릭재킹은 마우스 클릭을 하이재킹하여 사용자의 권한으로 특정 행동을 유도하는 공격 방법으로서, 온라인 계정을 삭제하거나 보인 기능 설정을 해제하는 등 여러 가지 문제점을 발생시킬 수 있다. 여러 가지 클릭재킹 방어 메커니즘이 제안되었으나 우회가 가능하여 잠재적인 문제점을 안고 있다. 따라서 본 논문에서는 현재 제안된 클릭재킹 방어기술들과 우회 기술들을 살펴보고 안전한 웹 브라우징 환경을 제공하는 브라우저를 제안한다.

1. 서론

인터넷 인프라가 발달하면서 인터넷 사용자가 증가하기 시작했고, 이에 따라 다양한 온라인 서비스를 제공되기 시작했다. 예를 들면, 과거에 오프라인으로 이루어졌던 은행 업무들이 모두 전산화되어 인터넷을 통해 고객들에게 제공되고 있으며 쇼핑 또한 온라인 상점들 간에 가격을 비교하여 최저가로 구매를 하면 구매자 앞으로 물건이 배송된다. 하지만 이러한 수익과 관련된 서비스들이 증가하면서 개인정보, 카드정보와 같은 민감한 정보들이 인터넷에 노출되기 시작했고, 이를 노린 사이버 범죄들이 등장하기 시작했다.



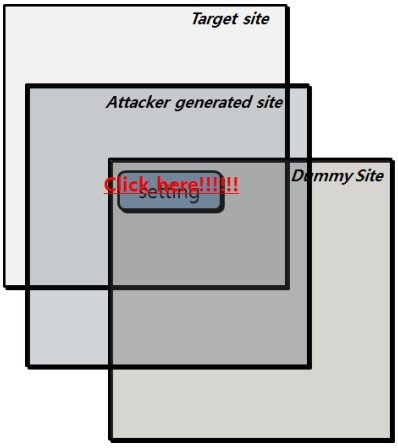
(그림 1) 농협 피싱 사이트[2]

공격자들은 결제정보를 탈취하기 위해, 그림1과 같이 은행 사이트와 똑같이 생긴 피싱 사이트를 만들어 사용자에게 보안카드번호 입력을 요구한다. 만약 피해자가 피싱 사이트를 구분 못하고 정보를 입력하게 되면, 해당 정보는 공격자에게 전송이 된다. 최근에는 단순한 피싱 사이트를 뛰어넘는 클릭재킹(Clickjacking)[1] 사이트들이 등장하기 시작했다. 기존의 피싱 사이트들이 단순하게 사회공학적으로 피해자에게 정보를 요구하여 탈취를 하였다면, 클릭재킹 사이트들은 피해자의 마우스 클릭을 유도하여 피해자의 권한으로 특정 행동을 하게 만든다. 공격자가 원하는 페이스북 게시글에 “좋아요” 버튼을 클릭하거나, 트위터 계정을 삭제하는 행동들이 대표적인 예이다. 현재 대부분의 웹 브라우저와 서버에서는 클릭재킹을 막기 위한 기술들을 적용하고 있지만 여전히 우회방법이 계속 등장을 하고 있고 클릭재킹의 위험은 줄어들지 않고 있다. 따라서 본 논문에서는 현재 사용되고 있는 클릭재킹 방어 기술들과 그 한계점들을 살펴보고, 그것들을 극복할 수 있는 웹브라우저를 제안한다.

본 논문의 구성은 다음과 같다. 2장에서 클릭재킹의 개념에 대해서 설명을 하고, 3장과 4장에서 클릭재킹 탐지기술 및 우회기술을 살펴본다. 5장에서는 본 논문에서 제안하는 웹브라우저의 전체 구조와 각각 모듈의 역할을 설명하고 6장에서는 제안한 웹브라우저의 한계점과 향후 연구방향을 제시하며 결론을 맺는다.

2. 클릭재킹

클릭재킹(Clickjacking)은 사용자를 속여 클릭을 유도함으로써 민감한 정보를 유출하거나 컴퓨터의 상태를 변경하는 공격 방법이다. 2002년에 투명 iframe이 보안 문제를 일으킬 수 있다는 점이 모질라 버그 트랙킹 시스템에 처음 개재되었으며 6년 뒤인 2008년에 Hansen과 Grossman이 개념을 정립하고 클릭재킹이란 용어를 정의하였다[1]. 그림2는 클릭재킹의 개념을 설명하고 있다.



(그림 2) 클릭재킹의 기본 개념

공격자는 사용자의 클릭을 유도할 웹페이지를 정해서 임의로 만든 공격 사이트 안에 프레임으로 삽입을 시킨다. 버튼의 위치 조정을 원활하게 하기 위해 추가적으로 더미 페이지를 만들어 왼쪽 상단에 프레임으로 삽입시킨 후 공격사이트로 재삽입 할 수도 있다. 피해자가 공격 사이트에 접속을 하여 특정 버튼이나 링크 등을 클릭하면, 동시에 타겟 사이트의 버튼이 클릭되는 효과를 나타낸다[1]. 이런 상황은 많은 문제들을 발생시킬 수 있다. 예를 들어, 사용자의 권한으로 클릭을 하기 때문에, 특정 보안 기능을 클릭재킹을 통해서 해제를 할 수도 있고, 피해자의 개인 정보를 다른 서버로 전송을 할 수도 있다.

3. 탐지 기술

3.1 노스크립트 클리어클릭(NoScript ClearClick)

NoScript[3]는 파이어폭스 웹브라우저의 확장기능으로서 사용자들에게 여러 가지 보안 기능을 제공한다. 예를 들어 특정 스크립트의 실행을 금지하거나 보다 안전한 프로토콜을 사용한다. 그 중 ClearClick[4]은 클릭재킹을 막는 기능으로서 실제로 사용자에게 보여지는 화면과 투명한 개체나 프레임을 시각화한 화면을 비교하여 만약 불일치 할 경우 클릭재킹 공격의 위험이 존재하므로 사용자에게 경고를 한다. 사용자는 그 경고를 보고 클릭을 할지 결정을 한다.

3.2 프레임 버스팅(Frame Busting)

Frame busting은 클릭재킹 공격을 방어하기 위해 제안된 초창기 기법들 중 하나로서, 인가되지 않은 프레임의 삽입을 막는 방법이다[5]. 관리자가 홈페이지에 프레임 버스팅 코드를 삽입 시키면 현재 페이지가 다른 웹 페이지의 프레임으로 삽입되지는 않았는지 판단을 한다. 아래는 frame busting 코드의 간단한 예이다.

```
if(top.location != location)
        top.location = self.location;
```

상위 프레임과 현재 프레임의 위치 값이 같지 않은 경우(다른 페이지에 프레임으로 삽입되어있을 경우) 상위 레이어의 위치 값을 삽입된 프레임의 위치 값으로 대체함으로써 프레임에서 벗어날 수 있다.

4. 우회 기술

4.1 더블 클릭재킹(Double Clickjacking)

더블 클릭재킹은 프레임을 만드는 대신 새로운 윈도우 창을 만들어 사용자의 클릭을 하이재킹하는 기술이다[6]. 공격자의 웹페이지에 접속하면 윈도우 창이 새로 생성되고 즉시 메인 윈도우창 뒤로 자신을 숨긴다. 그 후 사용자가 공격자의 웹페이지에 더블 클릭을 하게 되면(웹게임 등으로 유도) 첫 번째 클릭은 숨겨져 있던 창을 앞으로 띄우고, 두 번째 클릭은 공격자가 의도한 행동을 수행한다. 사용자는 이런 과정을 눈치 채지 못한다. 프레임을 페이지 내부에 삽입시키지 않았기 때문에 기존의 방법은 탐지를 하지 못한다.

4.2 네스티드 클릭재킹(Nested Clickjacking)

마이크로소프트사는 HTTP 헤더에 X-Frame-Options라는 헤더를 추가하여 클릭재킹 공격을 막는 방법을 제안하였다[7]. 헤더에는 DENY와 SAMEORIGIN 옵션이 있다. DENY 옵션을 보내면 프레임 내부의 렌더링을 거부하고, SAMEORIGIN 옵션의 경우 상위 프레임과 Origin이 동일한 경우에만 렌더링을 한다. 네스티드 클릭재킹 공격은 X-Frame-Options 헤더가 SAMEORIGIN으로 설정되어 있을 때 웹브라우저 페이지를 핸들링하는 과정에서 생기는 취약점을 이용한 우회 방법이다[8]. X-Frame Options가 SAMEORIGIN으로 설정이 되어있는 경우, 웹브라우져는 프레임된 페이지와 가장 상위 프레임만 비교하기 때문에 그 사이에 임의의 개수의 프레임이 존재하면 검사를 우회할 수 있다. 따라서 정상적인 웹페이지에 공격자의 사이트를 프레임으로 삽입시킬 수 있다면, 공격자의 사이트는 프레임 계층에서 중간에 존재하기 때문에 검사를 우회하여 클릭재킹을 할 수 있게 된다.

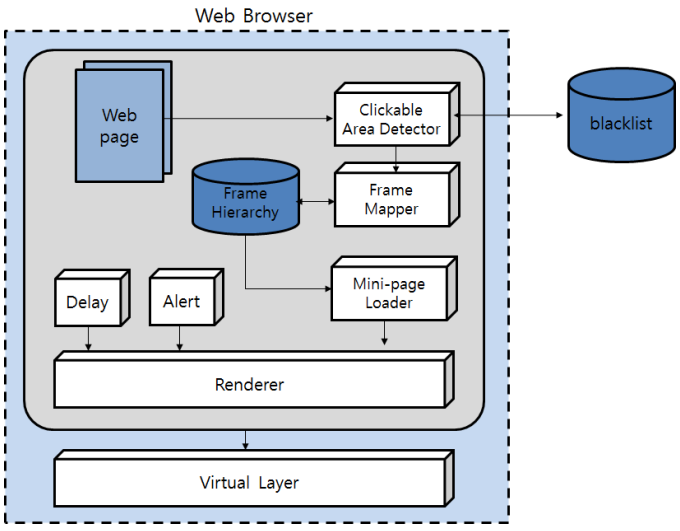
5. 클릭재킹 방지 웹브라우저

클릭재킹을 막기 위해서는 다음과 같은 요구조건들이 만족되어야 한다.

- 사용자는 마우스 클릭 대상을 시각적으로 확인할 수 있어야 한다.
- 프레임이 여러 계층으로 중첩되어 있어도 다른 도메인의 프레임을 식별할 수 있어야 한다.
- 다른 도메인에 접근 여부를 사용자가 선택할 수 있어야 한다.

클릭재킹은 사용자가 시각적으로 확인하지 못한 개체를 클릭함으로써 피해가 발생하는 공격 방법이기 때문에 일차적으로 사용자가 확인을 할 수 있게 해야 한다. 또한 현재의 보안책들은 단순히 2 계층의 프레임에 대해서만 방어를 하고 있기 때문에 여러 개의 프레임 계층이 존재할 경우 올바르게 대응할 수 없다. 마지막으로, 정상적인 웹페이지 또한 클릭재킹 사이트의 특징을 가질 가능성이 존재하기 때문에 마지막 판단은 사용자에게 의지를 해야 한다.

본 논문에서는, 위의 요구조건을 만족하는 웹 브라우저를 제안한다. 아래 그림은 제안하는 웹 브라우저의 전체 구조를 나타낸다.



(그림 3) 전체 구조

5.1 Clickable Area Detector

사용자들은 시각적으로 가려진 버튼을 클릭함으로써 피해를 입게 된다. 따라서 이 모듈은 페이지 내에 클릭이 가능한 구역을 식별하는 역할을 한다. 이 모듈에서 클릭 가능 구역을 Frame Mapper에 전달하면 프레임의 계층 구조를 구축하게 된다.

5.2 Frame Mapper

웹페이지 내에 삽입된 프레임들을 식별하는 모듈이다. Clickable Area Detector가 클릭 가능 구역을 전달하면, 해다 구역의 프레임의 계층 구조를 구축한다. 웹 페이지의 프레임 계층 구조는 트리 형태로 저장이 되며 노드에는 프레임의 도메인이 저장이 된다. 악성 도메인 데이터베이스와 트리의 노드들을 비교하여 만약 일치하는 도메인이 존재하면, 클릭재킹을 할 가능성이 높기 때문에 사용자에게 Alert 모듈을 통해 경고창을 띄운다.

5.3 Mini-page Loader

실제로 클릭 가능 구역에 마우스를 이동했을 경우 사용자에게 프레임 계층 구조를 인지시켜주는 모듈이며, 사용자가 보지 못하는 숨겨진 프레임 등을 미니맵 형태로 보여준다. 클릭을 하려 할 때, Frame Mapper에서 생성한 트리를 탐색하며 대상 프레임으로부터 일정 레벨의 중첩 프레임을 보여준다. 따라서 사용자가 버튼을 클릭하려고 할 때 실제 마우스 클릭 이벤트를 전달 받는 프레임이 무엇인지 시각적으로 보여주기 때문에 사용자는 미니맵을 보고 내가 클릭하고자 하는 도메인이 맞는지 손쉽게 판단을 할 수가 있다.

(그림 4) Mini-page Loader 예상 UI

5.4 Alert

사용자에게 경고창을 띄우기 위한 모듈이다. 과거에 클릭재킹을 했던 사이트들이 프레임이 삽입되어 있다든지, 아니면 현재 사용자가 공격에 노출되었을 때 사용자에게 이 사실을 인지시켜준다. 사용자는 경고창을 확인하고, 마우스 클릭의 승인 여부를 판단하면 된다.

5.5 Delay

클릭재킹 공격 중엔 투명한 프레임을 삽입하여 사용자의 클릭을 하이재킹하는 것들도 있지만, 사용자의 클릭 시점을 예상하여 매우 짧은 시간에 다른 페이지를 로드시키는 방법들도 있다. 이런 경우, 사용자는 A라는 버튼을 클릭하려고 했지만 클릭 순간에 페이지가 바뀌면서 버튼 B를 클릭하게 되는 것이다. 따라서 사람의 눈이 변화를 인지할 수 있는 시간까지 클릭을 금지시켜 클릭재킹 위협으

로부터 벗어날 수 있다. 이 아이디어는 2008년에 구글 엔지니어 Zalewski에 의해 제안되었다[8]. 새로운 페이지가 로드되면, 그 시간을 저장하여 마우스 클릭 이벤트가 발생할시 일정한 시간이 경과되었는지 확인하여 허용을 하거나 거부한다.

5.6 Virtual Layer

각 페이지끼리 서로 독립적으로 동작하기 위해 가상 레이어를 사용한다. 웹 페이지들은 서로 다른 인스턴스에서 구동이 되며 샌드박스 안에서 실행이 된다. 인스턴스의 생성과 소멸은 가상 레이어에서 관리를 한다. 악성 페이지가 웹브라우저의 취약점을 이용하여 로컬 컴퓨터에 접근을 하려 해도 샌드박스 내부이기 때문에 1차적인 방어가 되며, 가상 레이어의 기능을 확장하여 공격 탐지 모듈을 구현하면, 2차적인 방어를 할 수 있어 외부 공격으로부터 더 안전한 웹 브라우징 환경을 구축할 수 있다.

6. 결론 및 향후 연구

인터넷이 발달하고 사용자가 증가하면서 개인정보, 결제 정보 등을 탈취하기 위한 피싱 사이트들이 증가하기 시작했다. 더 나아가 사용자의 마우스 클릭을 하이재킹하여 웹캠, 녹음기를 실행시켜 사용자를 감시를 하는 사이트들도 등장하기 시작했다. 이런 공격을 클릭재킹(Clickjacking)이라고 부른다. 클릭재킹을 방어하기 위한 다양한 기술들이 제안되었으며 그들 중에는 실제로 구현이 되어 브라우저에 탑재된 것들도 있다. 하지만 보안 관련 연구원들에 의해 우회 방법이 발견되었으며 이것들이 만약 악용이 된다면 큰 문제를 발생시킬 수 있다. 따라서 본 논문에서는 Mini-page를 기반으로 한 클릭재킹 방지 웹브라우저를 제안하였다. 사용자가 웹 페이지를 방문할 때 숨겨진 프레임들을 찾아내서 그 부분으로 마우스를 이동했을 때 Mini-page를 사용자에게 보여주고 , 페이지 로드 후 마우스 클릭의 허용을 일정 시간 지연시켜 클릭재킹 공격을 방지한다.

하지만 본 논문에서 제안한 웹 브라우저는 몇 가지 한계점을 드러낸다. 첫째, 사용자의 이용성을 떨어트린다. 정상적인 웹페이지임에도 불구하고 탐지 조건을 만족하면 마우스 클릭을 지연시키거나 경고창을 띄우기 때문에 사용자 경험을 떨어트린다. 둘째, 악성 웹 페이지가 새로운 방법으로 클릭재킹을 시도할 경우, 현재 구조로서는 능동적으로 대처할 수 없다. 세 번째, 블랙리스트 데이터베이스를 정적으로 업데이트를 해야 한다. 브라우저는 접속한 사이트를 클릭재킹 사이트라고 판단을 하지 않고 사용자에게 경고만 하기 때문에, 동적으로 리스트를 업데이트 할 수 없기 때문이다.

향후에는 웹 페이지가 클릭재킹 공격을 하는지 여부를 자동적으로 판단하여 URL 블랙리스트 데이터베이스를 동적으로 관리하는 시스템을 연구할 계획이다. 데이터베이스가 자동으로 관리될 경우, 악성 사이트 정보가 빠른 속도로 퍼질 수 있어 클릭재킹 방지에 효과적일뿐더러, 클릭재킹 사이트 탐지 알고리즘은 따로 분리되어 다른 웹브라우저에 플러그인 형태로도 사용될 수 있기 때문에 클릭재킹 공격 방지의 관한 연구에 많은 기여를 할 것으로 기대한다.

Acknowledgement

본 논문은 지식경제부/산업기술평가관리원에서 지원하는 2012년도 산업원천기술개발사업(KI001810039260, 개인 및 기업 맞춤형 서비스를 위한 개방형 모바일 클라우드용 통합개발환경 및 이기종 단말-서버 간 협업 기술 개발)의 연구수행으로 인한 결과물임을 밝힙니다.

참고문헌

[1] Clickjacking 소개, http://www.sectheory.com/clickjacking.htm, Oct 2012
[2] 농협 피싱사이트, http://arumy.egloos.com/m/5624257, Oct 2012
[3] ClearClick, http://noscript.net/faq#clearclick, Oct 2012
[4] NoScript, http://noscript.net Oct 2012
[5] Gustav Rydstedt, Elie Bursztein, Dan Boneh, Collin Jackson, “Busting Frame Busting: a Study of Clickjacking Vulnerabilities on Popular Sites”, W2SP ‘10
[6] Lin-Shung Huang, Collin Jackson, “Clickjacking Attacks Unresolved”, CyLab July 2011
[7] Microsoft, “IE8 Security part vii: Clickjacking defenses“, 2009
[8] Sebastian Lekies, Mario Heiderich, Dennis Appelt, Thorsten Holz, Martin Johns, “On the fragility and limitations of current Browser-provided Clickjacking protection schemes”, USENIX WOOT’12 , Aug 2012
[8] M. Zalewski, “Dealing with UI redress vulnerabilities inherent to the current web”, http://lists.whatwg.org/pipermail/whatwg-whatwg.org/2008-September/016284.html, Oct 2012