Софийски университет "Св. Климент Охридски"
Факултет по математика и информатика

Реферат на тема On clickjacking attack

Основи на сигурно Уеб програмиране, летен семестър 2016/2017

спец. Софтуерно инженерство, 3 курс

Изготвила:
Елица Кехайова, Фак. №: 61837

Преподавател:

гл. ас. д-р Филип Петров

Предаване: ***Оценка:*……..
01.06.2017

София, 2017г.

Съдържание

Въведение

В момента Clickjacking атаката е голямо предизвикателство за експертите по сигурността и поради това незащитените уебсайтове се атакуват и се извършват измами. Тя е известна също като "атака срещу потребителския интерфейс". Изпълнява се, когато жертвата сърфира в уеб страницата, която вижда, но действията на кликването действително са скрити от атакуващият. Атакуващият използва множество прозрачни или непрозрачни слоеве, за да подмами жертвата да кликне върху бутон, поле за въвеждане или линк на страницата, която вижда. По този начин атакуващият "отвлича" кликванията и ги пренасочва към другата страница, която най-вероятно е собственост на друго приложение, домейн или и двете.

Според изследователят по сигурността Робърт Хансън има няколко варианта на Clickjacking. Някои от тях изискват достъп до различни домейни, други не. Едни от тях покриват цели страници, а други използват рамки за вграждане, за да ви накарат да кликнете върху едно място. Някои от тях изискват JavaScript, други- не изискват. Има варианти, които използват CSRF за предварително зареждане на данни във формуляри, а други не. Clickjacking не покрива нито един от тези случаи на употреба, а по-скоро всички тях.

Тази атака не използва софтуерни уязвимости в уеб приложенията, но се възползва от свойството HTML iFrame и от неговата непрозрачност. Това прави атаката доста статично нападение. Тя може да бъде използвана и за стартиране на други атаки, като Cross Site Request Forgery или Cross-Site Scripting.

Използвайки подобна техника, клавишите също могат да бъдат „отвлечени“. С внимателно изработена комбинация от стилове, рамки и текстови полета, жертвата може да бъде принудена да вярва, че въвежда паролата в имейла или банковата си сметка, но вместо това да пише в невидима рамка, контролирана от атакуващият.

През 2008 г. Робърт Хансен и Джеремай Гросман демонстрират за първи път Clickjacking Proof of Concept (POC), използвайки мениджъра на настройките на Adobe Flash Player, за да дадат на атакуващия достъп до камерата и микрофона на потребителя. Оттогава насам атаката Clickjacking е използвана при няколко широко разпространени атаки, включващи социални медии.

Има няколко налични метода за откриване или предотвратяване на атаки срещу „Clickjacking“ върху сървъра и клиента. Основните са за предотвратяване от страна на сървъра. Също така трябва да се проверява дали уебсайтът не се зарежда в iFrame. Ако атакуващият се опитва да зареди уебсайт с някой от тези методи, то той или ще "се счупи" извън рамката, т.е. да презарежда страницата директно към неговия URL адрес, или няма да се зарежда страницата.

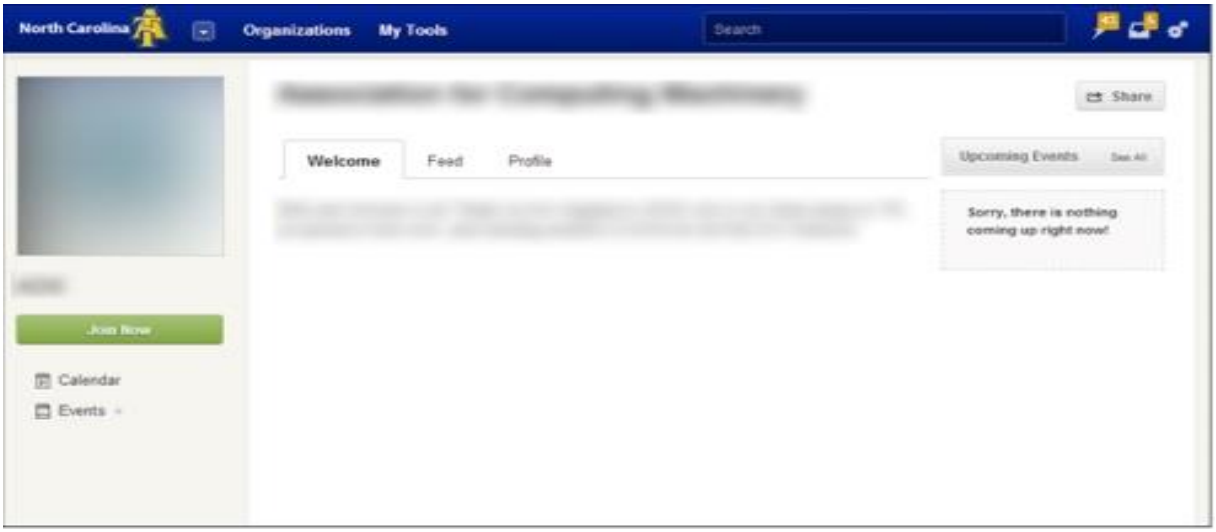
Как работи Clickjacking атаката

Атакуващият уебсайт зарежда целевия уебсайт в iFrame. За да гарантира правилното привеждане в съответствие с целевия уебсайт, нападателят може да използва серия iFrame-и с абсолютно позициониране или JavaScript, за да следи движението на мишката.

По-долу е даден пример за clickjacking, който използва серия iFrame-и с абсолютно позициониране. Кодът е базиран на пример в UI Redressing: Attacks and Countermeasures Revisited.

Фиг. 1 извежда изходния код за inner.html. Тя поставя целевата страница в iFrame, показана на Фиг. 1.1

Фиг.1 inner.html



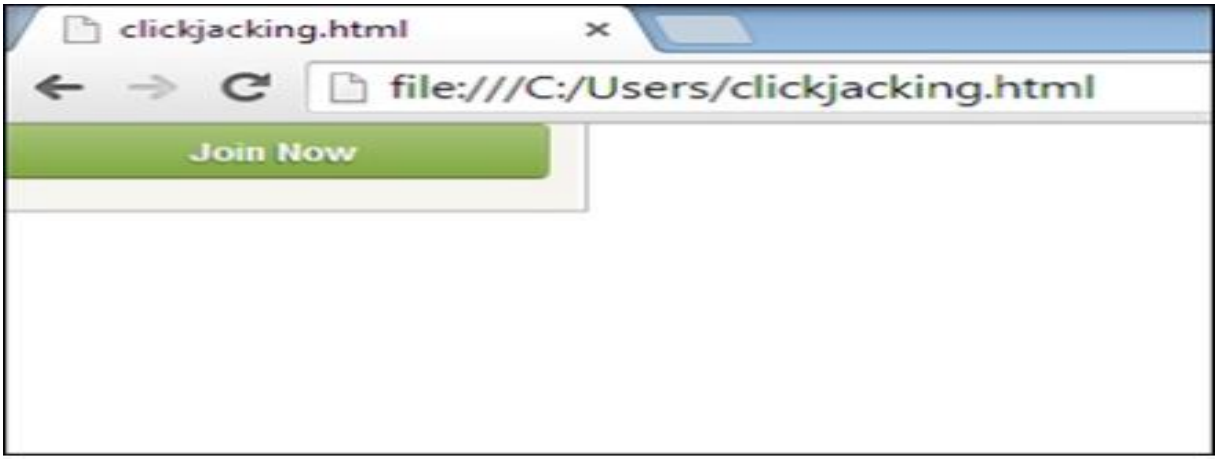
Фиг.1.1

Фиг. 2 извежда изходния код за clickjacking.html. Тя поставя inner.html в iFrame, което означава, че целевата страница вече е в две нива на iFrames. Крайният резултат е, че clickjacking.html показва само бутона Join Now вместо цялата страница, показана на Фиг.2.1.

clickjacking.html

```
1  <iframe id = "inner" src="inner.html" width="200" height="350" scrolling
   ="no" frameborder="none" style="position: absolute; left: -2"></iframe>
```

Фиг.2 clickjacking.html



Фиг. 2.1

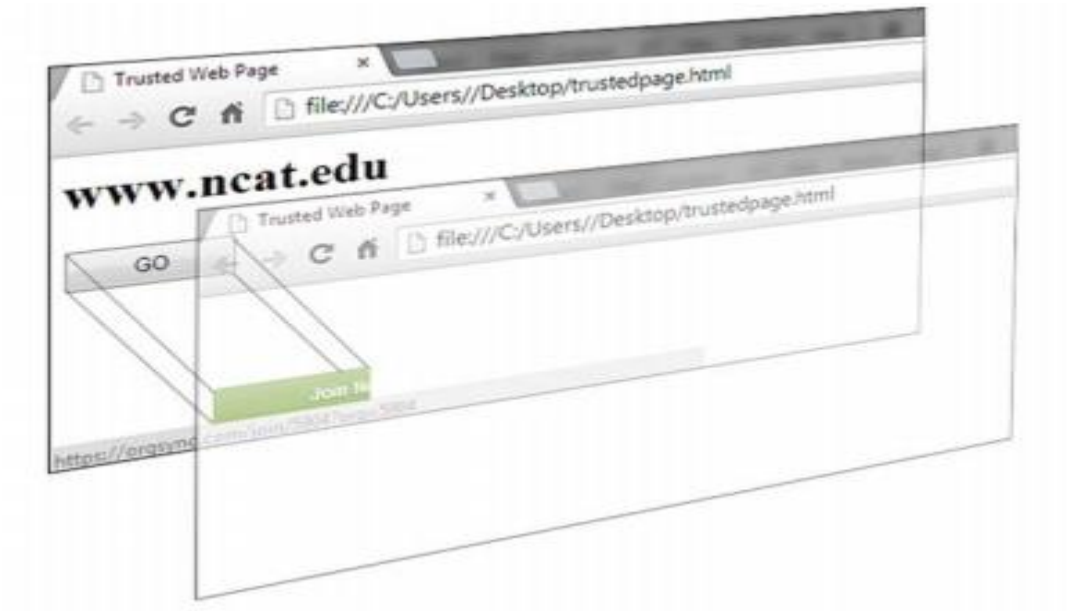
Фиг.3 извежда изходния код за trustedpage.html. Има iFrame на clickjacking.html, което означава, че целевата страница вече е в три нива на iFrames. Крайният резултат е, че trustedpage.html има бутон в точното местоположение, като бутона Join Now. Въпреки това, непрозрачността на iFrame беше зададена на 0.0, което означава, че бутонът Join Now на целевата страница е невидим, както е показано на Фиг.3.1. Фиг. 4 показва как iFrame-а покрива бутона GO. Ако жертва кликне върху бутона GO, то тя всъщност ще кликне върху бутона Join Now.

trustedpage.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/script.dtd">
<html>
<head>
    <title>Trusted Web Page</title>
</head>
<body>
    <h1>www.ncat.edu</h1>
    <form action="http://www.ncat.edu">
        <input type="submit" value="GO" style=" width: 100px; height:30
            px;">
    </form>
    <iframe id="clickjacking" src="clickjacking.html" width="97" height=
    "25" scrolling="no" frameborder="none" style="position: absolute;
        left: 9px; top: 68px; opacity: 0.0;"></iframe>
</body>
</html>
```

Фиг.3 trustedpage.html

Фиг. 3.1



Фиг. 4

Защо работи атаката Clickjacking

Повечето кликващи уебсайтове работят само ако жертвата вече е влязла в уебсайта - например сайт за социални контакти. Това позволява на атакуващия да подведе жертвата като извършва действията на доверен сайт и така вече тя да е удостоверена. Тъй като естеството на сайтовете за социални контакти може да се използва за бързо разпространяване на информация или връзки, както се вижда с вирусните видеоклипове, кликването често включва социална мрежа, за да се разпространи атаката.

Сайтовете за социални медии често предоставят бутони, които лесно могат да бъдат интегрирани, в която и да е уеб страница, например бутоните във Facebook "Like" или "Share", или бутоните в Twitter - "Tweet" и "Follow ".

Примери за Clickjacking атаки

През последните години са настъпили редица Clickjacking атаки в реалния живот. Някои от добре познатите са описани по-долу.

- Представете си нападател, който изгражда уеб сайт, върху него има бутон, който казва "кликнете тук за безплатен iPod". Но в горната част на тази уеб страница нападателят е заредил вградена рамка с мейла ви и е сложил бутона "изтриване на всички съобщения" директно над бутона "безплатен iPod". Жертвата се опитва да кликне върху бутона "безплатен iPod", но вместо това действително кликна върху невидимия бутон "изтрий всички съобщения". По този начин атакуващият "отвлича" кликването на потребителя.
- Един от най-известните примери за Clickjacking, е атака срещу страницата за настройки на приставката на Adobe Flash. Чрез зареждането на тази страница в невидима вградена рамка, атакуващият може да подмами потребителя да промени настройките за сигурност на Flash, давайки разрешение на всяка анимация на Flash да използва микрофона и камерата на компютъра му.
- Също така Clickjacking направи новини под формата на Twitter worm. Тази атака с clickjacking убеди потребителите да кликнат върху бутон, който ги накара да изпратят повторно местоположението на злонамерената страница и така да се разпространяват масово.

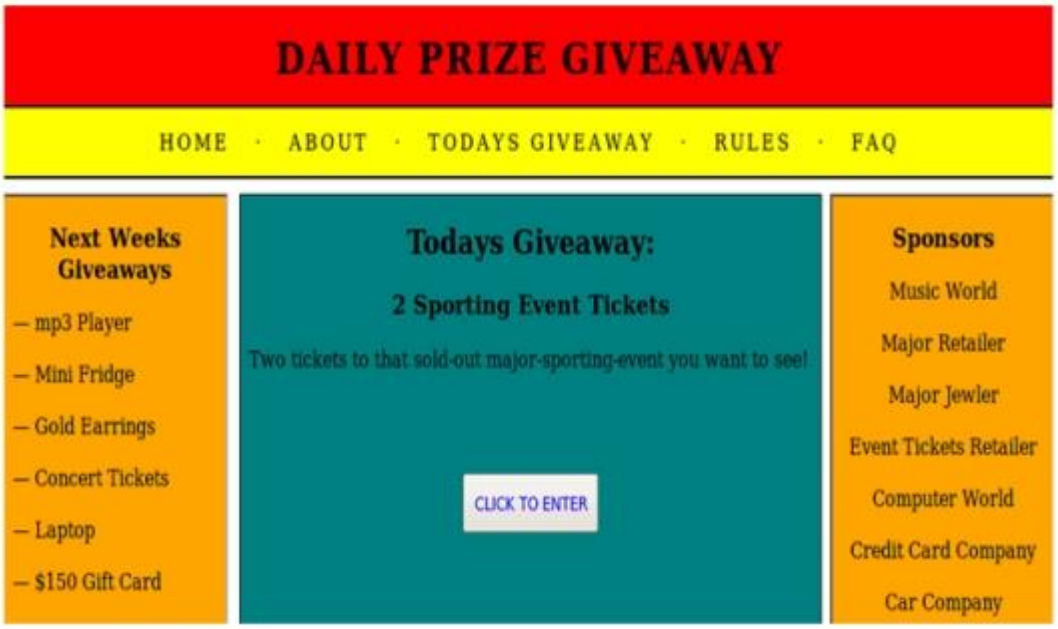
- Има и атаки с clickjacking, които злоупотребяват с бутона "Like" във Facebook. Атакуващите могат да измамят потребителите на Facebook, които са влезли в профила си, и така да харесат произволно фен страниците, връзките, групите и т.н.
- Атака срещу Facebook на тема Валентинки - През януари 2012 г. е направена Clickjacking атака във Facebook, където потребителите виждат публикация за инсталиране на тема "Валентинки" в своите Facebook акаунти. Ако потребителят(жертвата) последва връзката, е препращан до страница, на която те могат да инсталират темата "Свети Валентин" във своя Facebook.

  Ако потребителят кликне върху бутона "Инсталиране" и използва браузъра Chrome, бива подканен да изтегли файл с име FacebookChrome.crx и инсталира разширение на браузъра, наречено "Improvement". Също така автоматично харесва няколко страници във Facebook, както и автоматично публикува съобщение на стената си за засегнатите потребители. Изтегленият файл всъщност представлява злонамерен софтуер, който показва реклами.
- Атака срещу Twitter: „Don’t Click“ - Една от първите широко разпространени clickjacking атаки е " Don’t Click " в Twitter. Първоначално кодът бил разработен като POC, но всъщност някой го приложил и на практика. Самата атака не била злонамерена, но показала колко бързо може да се разпространи clickjacking атака в социалните медийни мрежи. Жертвите виждат връзка към съкратения URL-и в своите емисии, като например Фиг.5 и ако последват връзката, тя ги препраща на страница с бутон, който казва "Don’t Click me", както е показано на Фиг. 6. Кликването върху Бутонът кара жертвата да публикува краткия URL адрес на собствената си емисия, така че приятелите й да видят връзката.

Фиг. 5

Фиг.6

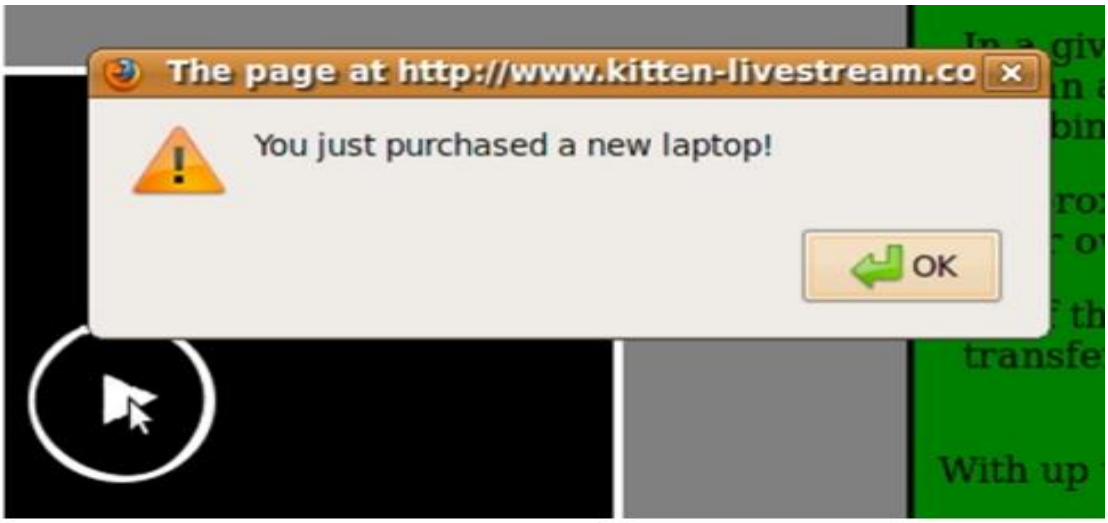
- Атака срещу уязвимостта на Chrome - През януари 2013 г. е открита уязвимост в браузъра Chrome, която би позволила на атакуващите да получат информация от жертвите им, която може да бъде страници на Google, Amazon, Microsoft Live и Yahoo! Profiles. Атаката включва "... двустранен метод на плъзгане и пускане, който се основава на това, че потребителите са подмамени да пуснат Chrome публично да публикува данните им".



Фиг. 7 Clickjacking атака, представляваща уебсайт на състезанието

Фиг. 8

Фиг. 9 Кутийка за предупреждение, която показва изтриването на електронна поща на жертвите

- 

Фиг. 10 Предупредителна кутия, представяща жертвата, която купува лаптоп от уебсайт за електронна търговия

- 

Фиг. 11 Представлява Don’t Click атаката в Twitter, където потребителят ще публикува връзка към емисията си

- Когато жертвата кликне върху връзката "PLAY A FREE ONLINE DEMO HERE" в уебсайта "нападател", трябва да се кликне върху бутона "Потвърди" на "легитимния" уебсайт.

Фиг. 12 Валиден уебсайт

Фиг. 13 Уебсайт „нападател“

- Clickjacker може да вземе бутон за вход от един сайт и да го скрие под друг елемент на невидима страница, която при кликване може да инициира злонамерен код.
- Потребителят може да получи имейл с връзка към видеоклип за дадена новина, но друга уеб страница, например продуктова страница на Amazon.com, може да бъде скрита отгоре или отдолу на бутона "PLAY" на новините. Когато потребителят се опита да "възпроизведе" видеоклипа, то всъщност "купува" продукта от Amazon.
- Атаката може да накара потребителите да направят обществената информация за профила си в мрежата.
- Сваляне и стартиране на злонамерен софтуер, който позволява на отдалечен хакер да поеме контрола върху другите компютри.
- Каране на потребителите следват някого в Twitter.
- Споделяне или харесване на връзки във Facebook.
- Да се хареса фен страница във Facebook или +1 в Google Plus.
- Кликвайки върху реклами от Google Adsense, за да генерират pay per click.
- Възпроизвеждане на видеоклипове в YouTube за печалба.
- Следвайте някого във Facebook.

- Likejacking е злонамерена техника за подлъгване на потребители в уебсайт да "харесат" на страница във Facebook, която не са имали намерение да "харесат".
- Cursorjacking е техника за поправяне на потребителския интерфейс за промяна на курсора от мястото, което потребителят вижда.

Макар че техническото изпълнение на тези атаки може да е предизвикателство поради несъвместимост между различните браузъри, редица инструменти като BeEF или Metasploit Project предлагат почти напълно автоматизирана експлоатация на клиенти на уязвими уебсайтове.

Методи за откриване и превенция

Като цяло тази онлайн атаката се развива бавно, така че е доста лесно да се предотврати даден уебсайт да е част от clickjacking атаката. Съществуват методи за превенция, както за клиента, така и за сървъра, както и метод за откриване от страна на клиента.

Откриване и предпазване от страна на клиента

Препоръчителният метод за откриване и предотвратяване от страна на клиента, е да се използва разширението NoScript на Firefox. То помага за предотвратяването на JavaScript базирани атаки, като блокира JavaScript, освен ако потребителят не го позволи да работи на даден уеб сайт, и включи модул, наречен ClearClick. ClearClick открива кликвания върху скрити елементи, припокриващи се с видими елементи, и сигнализира на потребителя,но въпреки това през 2010 г. проучване установи, че има много фалшиви положителни сигнали.

Екстремният метод за предотвратяване на clickjacking атаки е напълно да деактивирате JavaScript и iFrames в браузъра и да деактивирате всички приставки.

Атакуващите, които използват clickjacking често разчитат на жертва, която е влязла в уебсайт, като социална медийна мрежа, затова е препоръчително винаги да излизате от уебсайта.

Предотвратяване от страна на сървъра

- FrameBusting с JavaScript - От страна на сървъра, уебсайтовете могат да включват код, за да не бъдат уязвими на clickjacking атаки. Най-широко използваният метод е включването на JavaScript, за да се провери дали уебсайтът е бил поставен в iFrame, и ако е така, да се "счупи" извън рамката. Това се нарича FrameBusting.

  Този метод има недостатъци и често може да бъде заобиколен, като например се деактивира JavaScript-а в браузъра или в iFrame-а. Недостатъците на този метод включват затруднения, които допускат уебсайта ви да се оформя.

  OWASP препоръчва кода на JavaScript на Фиг. 14 за FrameBusting. Когато първата страница се зареди, всичко е невидимо. Ако JavaScript е активиран и страницата не е във вътрешността на iFrame, тя става видима. Ако JavaScript е активиран и страницата е във външността на iFrame, то тя се "чупи" от рамката, за презареждане на страницата директно и след това позволява съдържанието да бъде видимо. Това не се счита за силен метод за превенция, а е най-добрият метод за превенция на наследените браузъри.

index.html

```
<style id="antiClickjack">
    body{
        display: none !important;
    }
</style>
<style type="text/javascript">
    if (self===top){
        var antiClickjack = document.getElementById("antiClickjack");
        antiClickjack.parentNode.removeChild(antiClickjack);
    } else{
        top.location = self.location;
    }
</style>
```

Фиг. 14

- Изисква се JavaScript-а да бъде активиран –JavaScript-а се използва за важни action бутони за уебсайтовете, като submit бутони във form, това помага да се избегне заобикаляне на JavaScript FrameBusting.
- Валидиране извън рамките на групата - Валидирането на чувствителни действия, използващи комуникация извън обхвата на

групата, като електронна поща или SMS, гарантира, че действието ще се изпълнява само с познанията и съгласието на потребителите.

- HTTP Header - Най-добрият начин за предотвратяване създаването на рамка на уебсайт е да се включи X-FRAME-OPTIONS HTTP header. Когато е зададено на DENY, съдържанието не се зарежда, ако уеб страницата е в iFrame. Този метод е по-трудно да се заобиколи, въпреки че заглавието може да се премахне с помощта на прокси инструмент. Също така има опции, позволяващи на даден уебсайт лесно да се рамкира или да позволи на един домейн да зареди страницата в iFrame. Това е препоръчителният метод за превенция.

Защита срещу On Clickjacking атаки

Съществуват редица уязвимости в сигурността, които се експлоатират чрез използване на clickjacking атаки. Те варират от уязвимостите на Adobe Flash до опциите за управление на ActiveX. Този вид атака може да бъде трудна за контролиране, защото браузърът често вижда clickjacking атаките като оторизирани заявки от потребителя, като по този начин отваря пътя за извършване на всякакви злонамерени действия чрез браузъра на жертвата и чрез друг софтуер като Adobe Flash. Най-ефективните мерки за сигурност трябва да се направят на back-end, особено като се има предвид, че най-ефективното решение ще ограничи и възпрепятства функционалността на уебсайта.

Има два основни начина за предотвратяване на clickjacking атаките:

- Изпращане на правилен X-Frame-Options HTTP хедър на отговора, които оказва на браузъра да не допуска ферйминг от други домейни.
- Използване на защитен код в потребителския интерфейс, за да се гарантира, че текущият кадър е прозорецът с най-високо ниво.

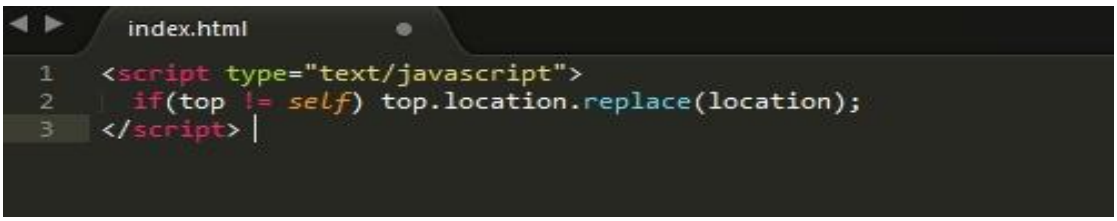
Други начини за предотвратяване са:

- Блокиране на скриптовете от браузъра - Един от начините за намаляване на риска е да се оценят и инсталират приставки за браузъри като NoScript и NotScript, които принуждават потребителите да разрешават действия на javascript на сайтовете, които посещават, както и да посочват доверени домейни.
- Internet Explorer - Internet Explorer 8 и по-новите имат някои предпазни мерки, които позволяват на уеб разработчиците да предотвратяват неоторизирани елементи на своите уеб сайтове. Това

означава, че уеб програмистът може да защити собствената си страница от злонамерен код, който може да дойде от вградени реклами или друго съдържание.

- Firefox – Инсталиране на приставките за NoScript за Firefox. NoScript ще предотврати възпроизвеждането на всички видеоклипове с Flash, когато посетите сайт чрез NoScript, след като блокира цялото съдържание на Flash, автоматично ще блокира и Flash рекламите.

  NoScript - блокира JavaScript, Java, Flash, Silverlight и друго "активно" съдържание по подразбиране във Firefox. Това е базирано на предположението, че злонамерените уеб сайтове могат да използват тези технологии по вредни начини. Потребителите могат да разрешат активното съдържание да се изпълнява на доверени уеб сайтове.
- Framekiller (framebuster или framebreaker) е част от JavaScriptcode, която не позволява на уеб страниците да се показват в рамка. Рамката е подотдел на прозореца на уеб браузъра и може да действа като по-малък прозорец. Този вид скрипт често се използва за предотвратяване на дадена рамка от външен уебсайт да се зарежда от рамките без разрешение. Типичният изходен код за скрипта framekiller е:



- Window.confirm() защита - Използването на x-frame- options или frame-breaking скрипт е по-безотказан метод за защита от clickjacking атаки. Въпреки това, при сценарии, при които съдържанието трябва да е подлежащо на разглеждане, може да се използва window.confirm(), за да се облекчи clickjacking атаката, като се информира потребителя за действието, което ще извърши.

  Извикването на window.confirm() ще покаже изскачащ прозорец, който не може да бъде оформен. Ако window.confirm() произхожда от вградена рамка с различен домейн от родителския, диалоговият прозорец ще покаже кой домейн е произхождал от window.confirm (). В този сценарий браузърът показва произхода на диалоговия прозорец, за да помогне за смекчаване на clickjacking атаките. Трябва да се отбележи, че Internet Explorer е единственият

известен браузър, който не показва домейна от диалоговия прозорец window.confirm(), за да се реши проблема с Internet Explorer, трябва да се уверите, че съобщението в диалоговия прозорец съдържа контекстната информация за вида на действието, което се изпълнява. Например:

index.html

```
<script type="text/javascript">
    var action_confirm = window.confirm("Are you sure you want to delete
        your youtube account?")
    if (action_confirm) {
        //... perform action
    } else {
        //...  The user does not want to perform the requested action.
    }
</script>
```

- The onBeforeUnload Event - Потребителят може ръчно да анулира заявката за навигация, подадена от рамковата страница. За да се възползва от това, рамковата страница регистрира onBeforeUnload handler, който е наречен рамкова страница и е на път да бъде незаредена поради навигацията. Функцията "handler" връща низ, който става част от подканата, показвана на потребителя. Например, нападателят иска да очертае PayPal. Той регистрира незареденатa handler функция, която връща низа "Искате ли да излезете от PayPal?". Когато този низ се покаже на потребителя, е вероятно той да прекрати навигацията, като попречи на опита за изваждане на рамката на PayPal. Нападателят извежда тази атака, като регистрира събитие за незареждане на горната страница, като използва следния код:

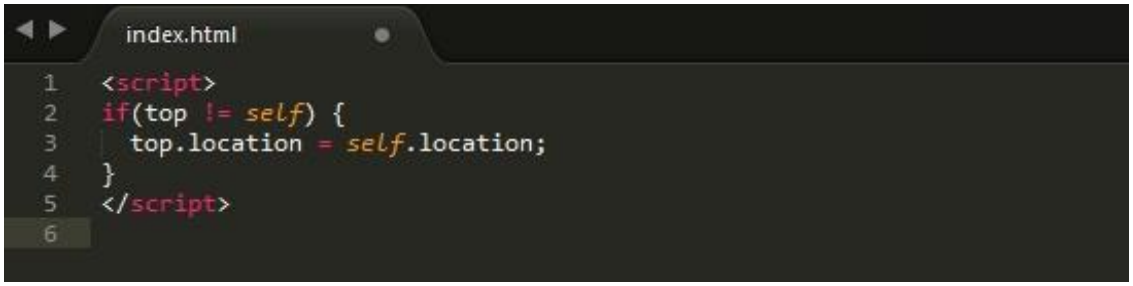
index.html

```
<script>
window.onbeforeunload = function()
{
  return "Asking the user nicely";
}
</script>
<iframe src="http://www.paypal.com">
```

Кодът за изтриване на кадрите на PayPal ще генерира BeforeUnload събитие, което ще активира функцията и ще прикани потребителя да анулира навигационното събитие.

- Използване на XSS филтри - IE8 и Google Chrome въведоха отразяващи XSS филтри, които помагат за защитата на уеб страници от определени видове атаки на XSS. Филтърът XSS в IE8 сравнява дадените параметри на заявката с набор от регулярни изрази, за да търси очевидни опити за cross-site scripting. Използвайки "induced false positives", филтърът може да се използва за деактивиране на избраните скриптове, като съчетава началото, на който и да е скриптов маркер в параметрите на заявката. Филтърът XSS ще деактивира всички вградени скриптове в страницата, включително скриптовете за премахване на рамки. Външните скриптове могат да бъдат насочени и чрез съвпадение на външно включване, което ефективно изключва всички външни скриптове. Тъй като подмножествата на заредения JavaScript са все още функционални (вградени или външни) и "бисквитките" все още са налице, тази атака е ефективна при кликване.

  Код за изваждане на рамката на жертвите:



Нападател:

XSS филтърът ще съответства на този параметър "<script>, if" до началото на frame busting скрипта за изваждане на кадъра на жертвата, и следователно ще забрани всички вградени скриптове в страницата на жертвата, включително скрипта за изваждане на рамката. Филтърът XSSAuditor, достъпен за Google Chrome, дава възможност за едно и също използване.

Тенденции на Clickjacking атаките

През август 2011 г. Symantec публикува проучване от 3,5 милиона публикации във Facebook и "... установи, че до 15% от уникалните публикации са били идентифицирани като likejacking атаки".

Bitdefender съобщи за едно случайно проучване по същото време на измамата "Вижте кой видя вашия профил", което разкри приблизително 1,4 милиона кликвания, генерирани по време на атаката. Тя се отрази 34 часа, след като връзката се разпространи.

Използвана литература

[1] https://www.owasp.org/index.php/Clickjacking
[2] http://socio.org.uk/isej/fulltext/v1n2/4.pdf
[3] https://www.symantec.com/connect/articles/short-information-clickjacking-attacks?page=1
[4] https://www.owasp.org/index.php/Clickjacking_Defense_Cheat_Sheet
[5] https://en.wikipedia.org/wiki/Clickjacking