A cosa servono gli attributi noopener e noreferrer?

sabato, 06 maggio 2017

Quante volte hai visto link aperti in un nuovo pannello (oppure in una nuova finestra) del browser sui siti web che visiti, o magari li hai personalmente inseriti nei siti che produci?
Questo tipo di collegamento, si ottiene attraverso l'attributo specifico *target="_blank"*, lo saprai di certo. Ma sapevi che i link con tale attributo espongono gli utenti a possibili tentativi di phishing?
Ma procediamo con ordine e proviamo a comprendere l'argomento.

Sommario

In cosa consiste la vulnerabilità?

Ebbene sì, la banale apertura di un link in *target="_blank"* espone l'utente che compie l'azione a potenziali pericoli. Ma come? Non tutti sanno che la pagina aperta attraverso un link, ha una relazione con la pagina di provenienza, relazione che può essere sfruttata per avere parziale accesso a quest'ultima.

La proprietà "opener" di JavaScript

JavaScript ha tra le sue proprietà una in particolare, chiamata "opener" che contiene appunto un riferimento alla pagina di provenienza. Ma passiamo alla pratica. Prova a creare due semplici pagine:

- index.html
- target.html

In index.html, inserisci un semplice link alla pagina target.html:

```
<!DOCTYPE html>
<html lang="it">
<head>
 <meta charset="UTF-8">
 <title>PAGINA 1</title>
</head>
<body>
 <h1>Questa Ã¨ la pagina di provenienza</h1>
 <a href="target.html" target="_blank">vai alla pagina target</a>
</body>
</html>
```

mentre in target.html, inserisci il seguente codice:

```
<!DOCTYPE html>
<html lang="it">
<head>
 <meta charset="UTF-8">
 <title>PAGINA TARGET</title>
</head>
<body>
 <script>console.log( window.opener );</script>
</body>
</html>
```

Aprendo index.html e cliccando sul link, vedrai, come previsto, aprirsi un nuovo pannello nel tuo browser, contenente la pagina target.html.
Fin qui tutto normale, ma prova ad aprire la console JavaScript del tuo browser in corrispondenza della pagina appena aperta:

Non sai come fare?

Chrome

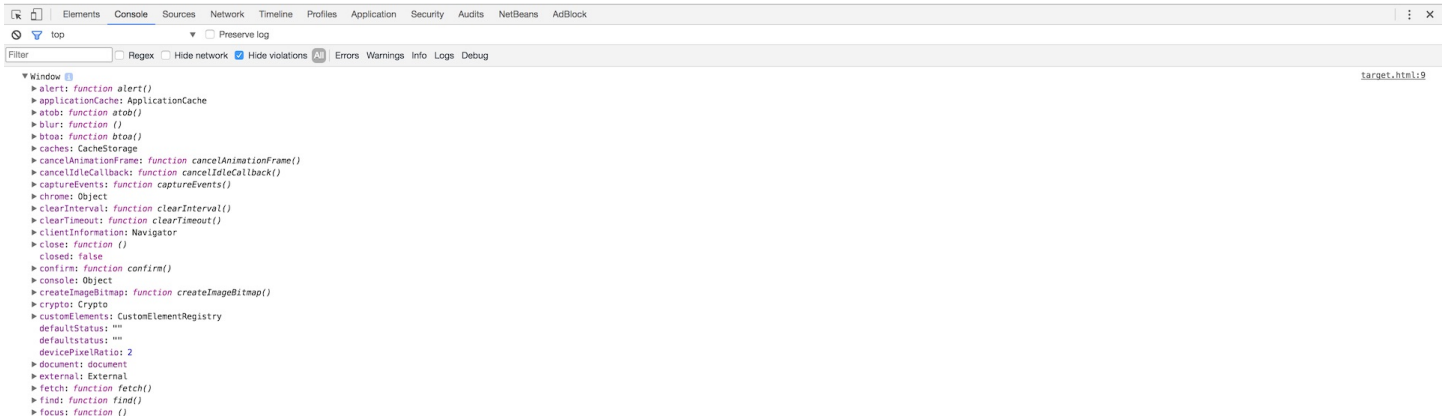
- Ctrl+Shift+J su Windows/linux
- Cmd+Opt+J su MacOsX

Firefox

- Ctrl+Shift+K su Windows/linux
- Cmd+Opt+K su MacOsX

Vedrai la presenza dell'oggetto *Window*, che è appunto l'oggetto contente tutte le informazioni relative alla pagina di provenienza (index.html).

Questa è la pagina di atterraggio



Un esempio di phishing

Bene, ora sappiamo e abbiamo testato che nella pagina di atterraggio, attraverso JavaScript, abbiamo a disposizione una serie di proprietà della pagina di provenienza, su cui naturalmente possiamo agire e modificare.

Aggiungi alla pagina target.html il seguente codice:

```
<script>window.opener.location = 'https://www.google.it';</script>
```

Ma cosa significa il codice inserito? Abbiamo agito sulla proprietà location dell'oggetto contenente i dati della pagina di provenienza, a cui diciamo di reindirizzare su *www.google.it*. In effetti, tornando al pannello che conteneva index.html, vedrai che adesso è puntata su *www.google.it* !!

Come può essere sfruttata la situazione?

Ipotizziamo che un malintenzionato crei un post virale ad esempio su Facebook, con link alla pagina target.html.
Da premettere che Facebook, per sua natura, apre i link dei post in *target="_blank"*.
Cosa succederebbe se il link contenuto nella proprietà window.opener.location indirizzasse, anziché *www.google.it*, ad una pagina che esteticamente è del tutto uguale a quella di login di Facebook?
Beh, facile intuirlo! Molti, moltissimi utenti poco smaliziati, potrebbero cadere nel tranello ed inserire in questa pagina le proprie credenziali di accesso... e la frittata è fatta!

L'attributo "rel"

Come chiudere questa falla?
La prima cosa da fare, è prendere confidenza con l'attributo *rel*, specifico del tag HTML *<a>*.
L'attributo *rel*, che può essere utilizzato solo con il tag *<a>*, definisce la relazione tra la pagina HTML attuale e la pagina HTML linkata nel tag *<a>*. Può assumere diversi valori, forse uno dei più conosciuti ed utilizzati è *nofollow*, il quale indica che l'attendibilità della pagina linkata non è in qualche modo comprovata, in quanto si tratta magari di un link esterno al nostro sito.
In particolare, ad esempio, l'attributo *rel="nofollow"* viene utilizzato da Google per specificare che lo spider non deve "seguire" quel link, in alcuni casi, appunto perché magari esterno.

Noopener, Noreferrer

Tra i valori possibili dell'attributo *rel*, ci sono appunto *noopener noreferrer* (**noopener** per Google Chrome, **noreferrer** per Firefox), che bloccano la trasmissione dell'oggetto JavaScript *Window*, precedentemente trattato, che fa riferimento alla pagina di provenienza.
Prova infatti a modificare la pagina index.html in questo modo:

```
<!DOCTYPE html>
<html lang="it">
<head>
 <meta charset="UTF-8">
 <title>PAGINA 1</title>
</head>
<body>
 <h1>Questa Ã¨ la pagina di provenienza</h1>
 <a href="target.html" target="_blank" rel="noopener noreferrer">vai alla pagina
target</a>
</body>
</html>
```

quindi apri la pagina e clicca sul link... cosa è cambiato?
La pagina non viene più reindirizzata a *www.google.it*, pur aprendo come previsto la pagina target.html in un nuovo pannello del browser.

Conclusioni

Attualmente con la versione 4.7.4, WordPress ad esempio inserisce di default l'attributo *rel="noopener noreferrer"* nei link in *target="_blank"* inseriti attraverso l'editor TinyMCE nel backend. È dunque una buona pratica inserire l'attributo *rel="noopener noreferrer"* nel tuoi link in *target="_blank"*?

La risposta non è semplice ed infatti scatena numerose discussioni sul web riguardanti eventuali "controindicazioni" circa la SEO, in particolare per i backlink, ma non sembra esserci un riscontro diretto e dimostrato sull'argomento, per ora.

Se vuoi saperne di più ti consiglio di leggere le discussione in merito:
Wordpress Trac Ticket

Ed anche l'ulteriore discussione sulla community di Moz, in cui si smentiscono le eventuali problematiche SEO derivanti dall'utilizzo di *rel="noopener noreferrer"*:
Moz community discussion

Ma se comunque preferisci non rischiare in attesa che si abbia più chiarezza sull'argomento, ecco come disabilitare l'inserimento automatico dell'attributo *rel="noopener noreferrer"* nei link esterni inseriti tramite l'editor di WordPress (a partire dalla v4.7.4):
Disabilitare l’inserimento di noopener e noreferrer in WordPress