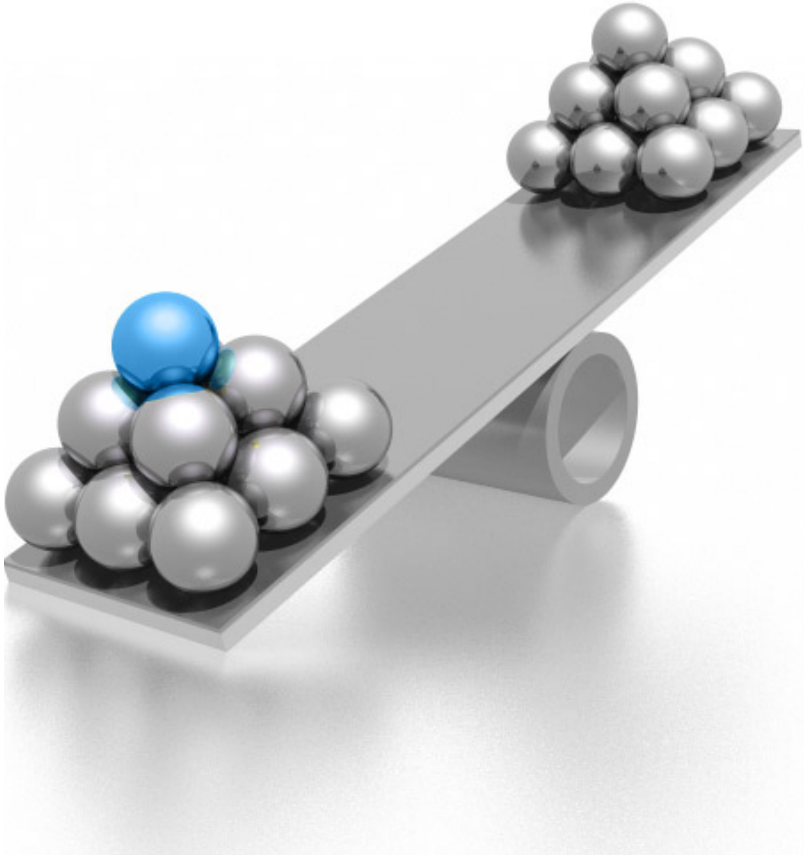
ICEBERG CAPITAL PARTNERS

Aaron Emigh. Inventor.

Patent Acquisition Opportunity

Strictly confidential
June 2015

Table of Contents



Executive Summary

Portfolio overview

- The patent offering from X comprises 5 US patents and 1 US application across 4 families.
- There are 3 Key Patents in this portfolio.
- The technologies in this portfolio relate to security for documents to ensure the integrity of the documents exchanged over a network.
- Notable forward citing companies include Red hat, Facebook, Google, and Microsoft.
- The earliest priority date in the portfolio is 4th February 2004.

Encumbrances

- No licenses.
- No buyer restrictions.

Transaction Profile

- ICEBERG Role: Sell-side adviser.
- Guide price:
  US $1.8m for entire portfolio.
- Grantback license required.
- Indication of interest requested to be submitted by: 14th August 2015.

Appendix

- Evidence of Use analysis suggesting infringement by Google and Microsoft, and further mapping against W3C standards.

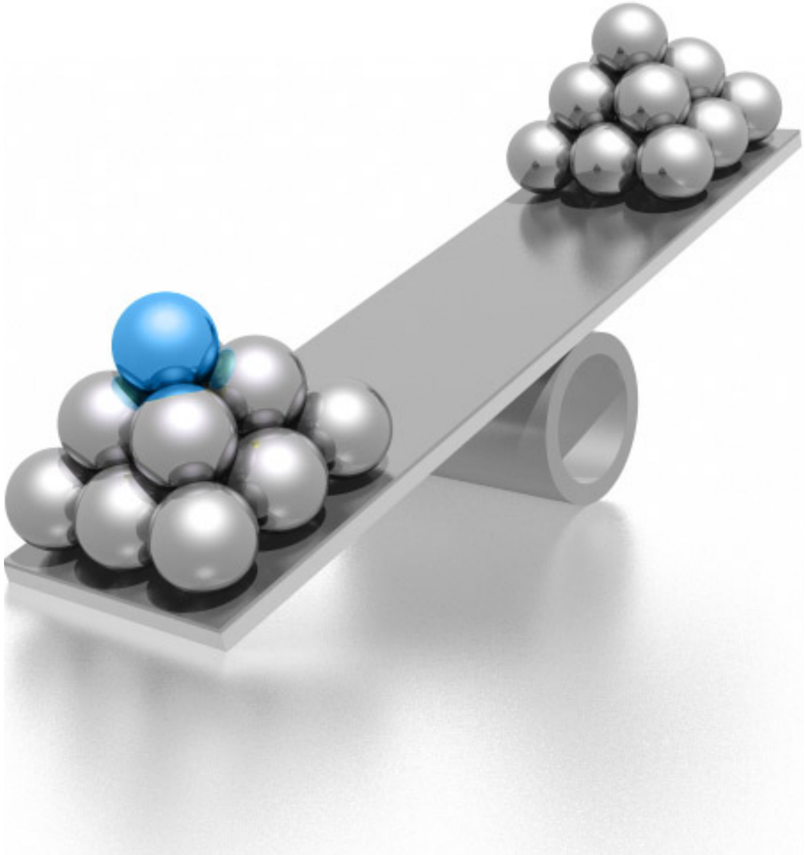
Seller Information

Based in Silicon Valley, Aaron Emigh has been founder, CTO, and CEO of several companies; including shopkick, Six Apart, Rojo Networks, and CommerceFlow; a company based on the ultrasonic signaling technology invented by Mr. Emigh. In 2003 CommerceFlow was sold to eBay, and in 2014 he sold shopkick for US $250m.

Over the past 30 years, Mr. Emigh has led technology initiatives in many fields, including security, anti-spam, anti-phishing, mobile, social media, e-commerce, data compression, multimedia, vision, machine learning, networking, and storage. His work in these fields has resulted in over 65 issued U.S. patents as well as successful business outcomes. Mr. Emigh is a frequent public speaker on technology and entrepreneurial topics.

Mr. Emigh has been recognized by the World Economic Forum as a Technology Pioneer, and has served as a member of the US Secret Service Electronic Crimes Task Force and the US-DHS Infosec Technology Transition Council, a Research Fellow of the Anti-Phishing Working Group, and a technical advisor to the Financial Services Technology Consortium.

Location: USA

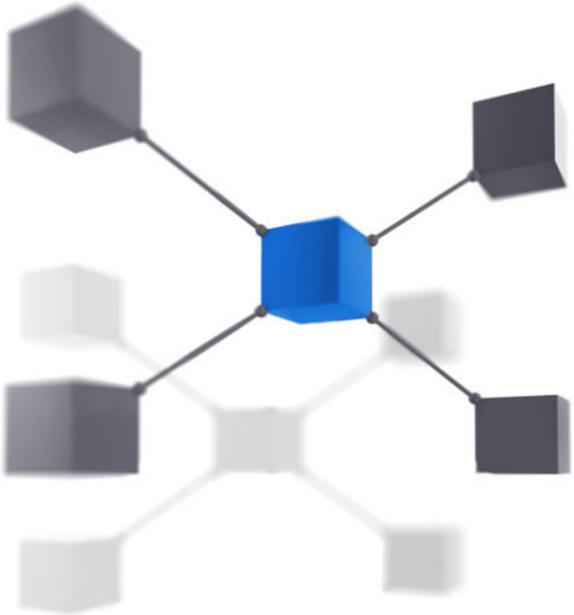


Key Patents

US 8104092 – Relates to website content security policies that are designed to prevent cross-site scripting attacks and other related vulnerabilities. The technology described in the patent delivers security benefits to developers to protect their website scripts and document integrity.

US 7712142 – Relates to website content security policies that are designed to prevent cross-site scripting attacks and other related vulnerabilities. The technology described in the patent delivers security benefits to developers to protect their website scripts and document integrity.

US 8423471 – Relates to the functionality of a web browser. Specifically the patent describes the detection of a request to traverse a hyperlink included in the web page. The request is made by the user when the user clicks the hyperlink. The patent describes how the browser evaluates the traverse request determining whether to traverse the link based on the evaluation.



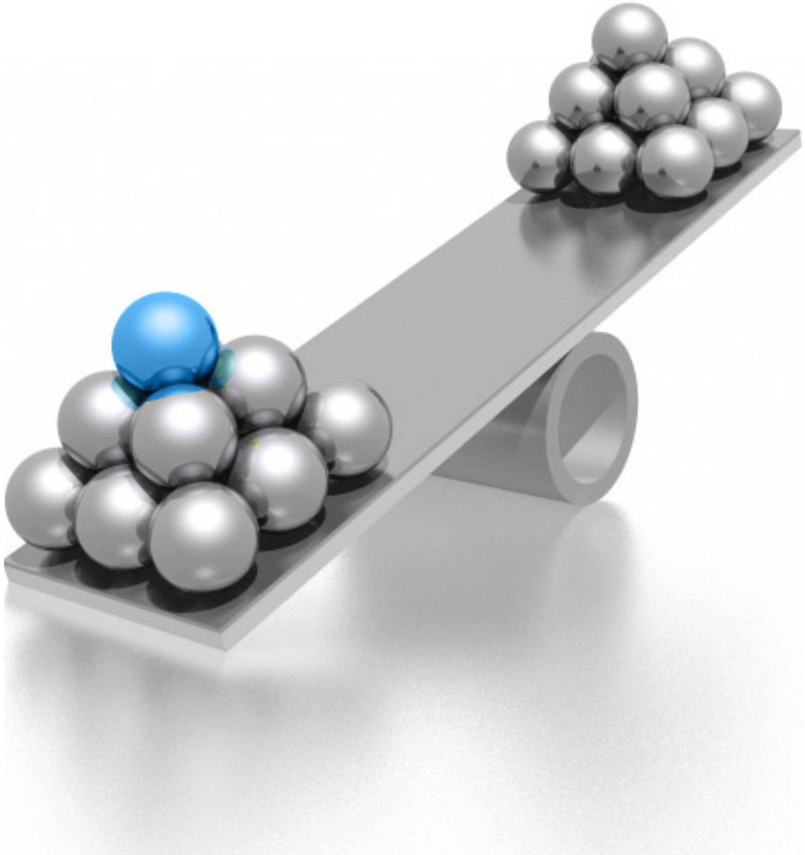
Patent List

Family	US Patent	US	Priority	Title
1	**8104092***	●	**10/09/2005**	**Document integrity assurance**
1	**7712142***	●	**10/09/2005**	**Document integrity**
2	8645480	●	19/07/2009	Trust representation by similarity
3	**8423471***	●	**04/02/2004**	**Protected document elements**
3	13/856036	●	04/02/2004	Enforcement of Document Element Immutability
4	8965892	●	04/01/2007	Identity-based filtering

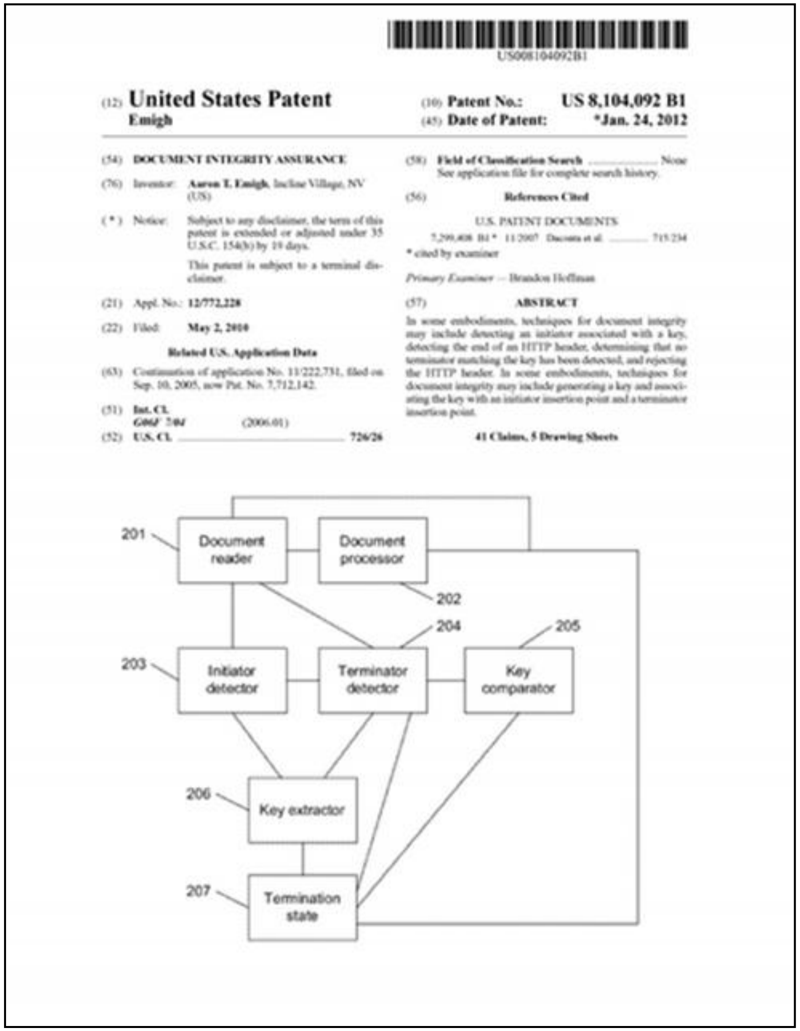
* **Key Patent** – see Appendix

Appendix

Evidence of Use



Family 1: US 8104092 – Bibliographic information



US008104092B1

(12) **United States Patent**
Emigh

(10) **Patent No.:** **US 8,104,092 B1**
(45) **Date of Patent:** ***Jan. 24, 2012**

(54) **DOCUMENT INTEGRITY ASSURANCE**

(76) Inventor: **Aaron T. Emigh**, Incline Village, NV (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 19 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **12/772,228**

(22) Filed: **May 2, 2010**

Related U.S. Application Data

(63) Continuation of application No. 11/222,731, filed on Sep. 10, 2005, now Pat. No. 7,712,142.

(51) **Int. Cl.**
G06F 7/04 (2006.01)

(52) **U.S. Cl.** **726/26**

(58) **Field of Classification Search** .... None
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

7,299,408 B1 * 11/2007 Dacosta et al. 715/234

* cited by examiner

Primary Examiner — Brandon Hoffman

(57) **ABSTRACT**

In some embodiments, techniques for document integrity may include detecting an initiator associated with a key, detecting the end of an HTTP header, determining that no terminator matching the key has been detected, and rejecting the HTTP header. In some embodiments, techniques for document integrity may include generating a key and associating the key with an initiator insertion point and a terminator insertion point.

41 Claims, 5 Drawing Sheets

Patent of Interest:

US8104092
(Priority date: Sept 10, 2005)

Document Integrity assurance

Exemplary Market Applications:

The patented technology finds applications in document security.

US 8104092 – Claim 10

10. A method for document integrity, comprising:

detecting an initiator insertion point in an electronic document;

generating a key;

associating the key with an initiator at the initiator insertion point;

associating the key with a terminator at a terminator insertion point, wherein the terminator corresponds to the initiator; and

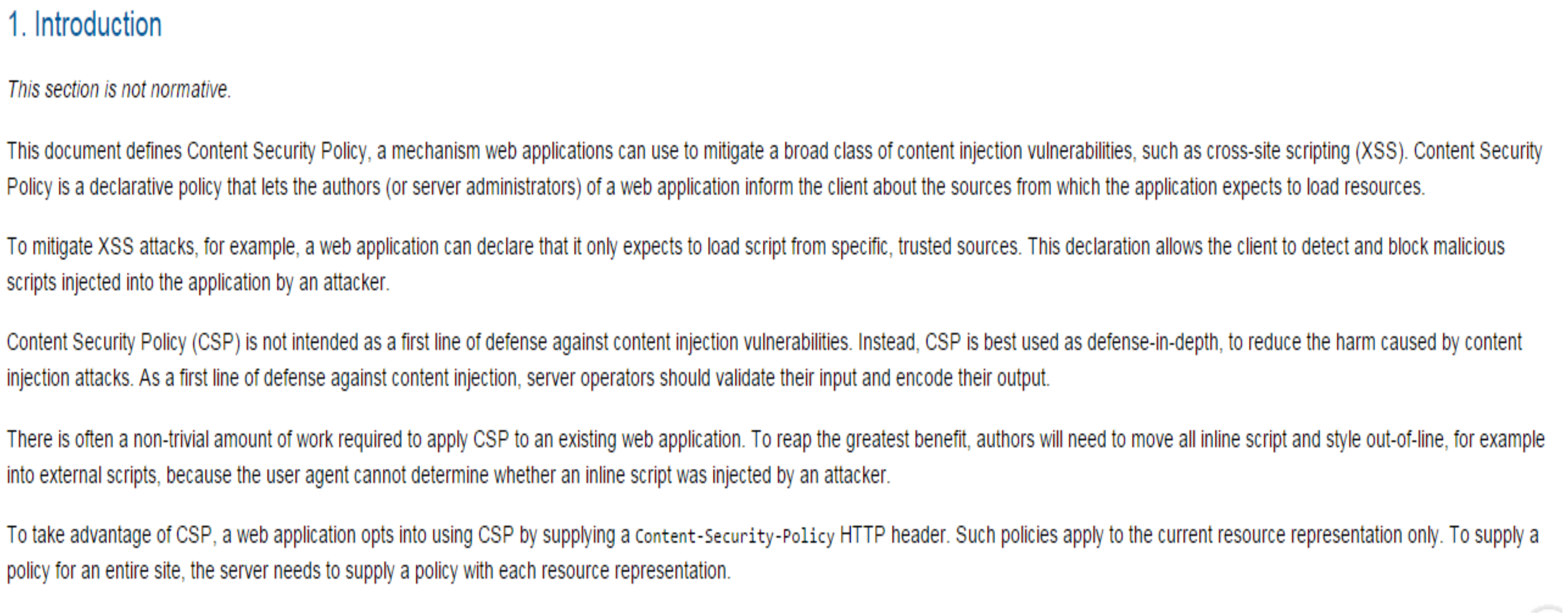
saving the document.

US 8104092 – EoU Summary

W3C	
Key claim(s)	1, 6, 10, 21, 26, and 34 (Independent Claims)
Mapped product	Chart has been made with respect to W3C. http://www.w3.org/TR/CSP/
Source	Information related to product is available at: http://www.w3.org/TR/CSP/
Product launch date	Content Security Policy was launched in 2012
Details of standard	The World Wide Consortium (W3C) is an international standard organization for World Wide Web. The Candidate Recommendation of this organization includes a policy called Content Security Policy. This policy is for computer security as it prevents attacks like cross-site scripting.

US 8104092 – W3C: Overview

The World Wide Consortium (W3C) is an international standard organization for World Wide Web. The Candidate Recommendation of this organization includes a policy called Content Security Policy. This policy is for computer security as it prevents attacks like cross-site scripting.



1. Introduction

This section is not normative.

This document defines Content Security Policy, a mechanism web applications can use to mitigate a broad class of content injection vulnerabilities, such as cross-site scripting (XSS). Content Security Policy is a declarative policy that lets the authors (or server administrators) of a web application inform the client about the sources from which the application expects to load resources.

To mitigate XSS attacks, for example, a web application can declare that it only expects to load script from specific, trusted sources. This declaration allows the client to detect and block malicious scripts injected into the application by an attacker.

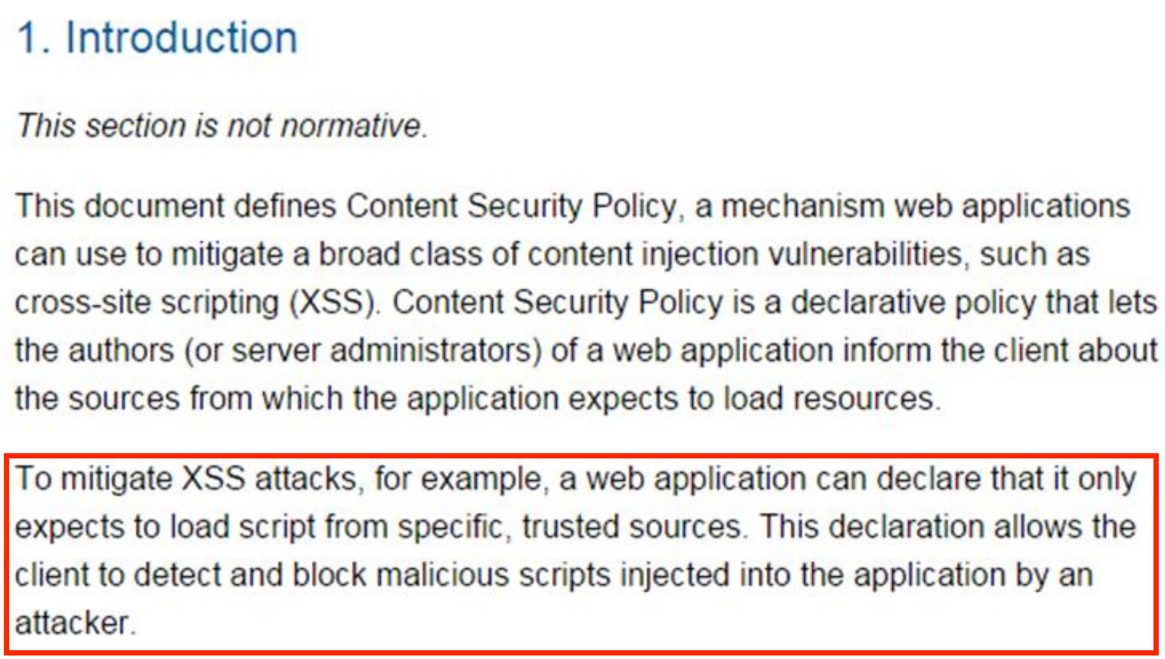
Content Security Policy (CSP) is not intended as a first line of defense against content injection vulnerabilities. Instead, CSP is best used as defense-in-depth, to reduce the harm caused by content injection attacks. As a first line of defense against content injection, server operators should validate their input and encode their output.

There is often a non-trivial amount of work required to apply CSP to an existing web application. To reap the greatest benefit, authors will need to move all inline script and style out-of-line, for example into external scripts, because the user agent cannot determine whether an inline script was injected by an attacker.

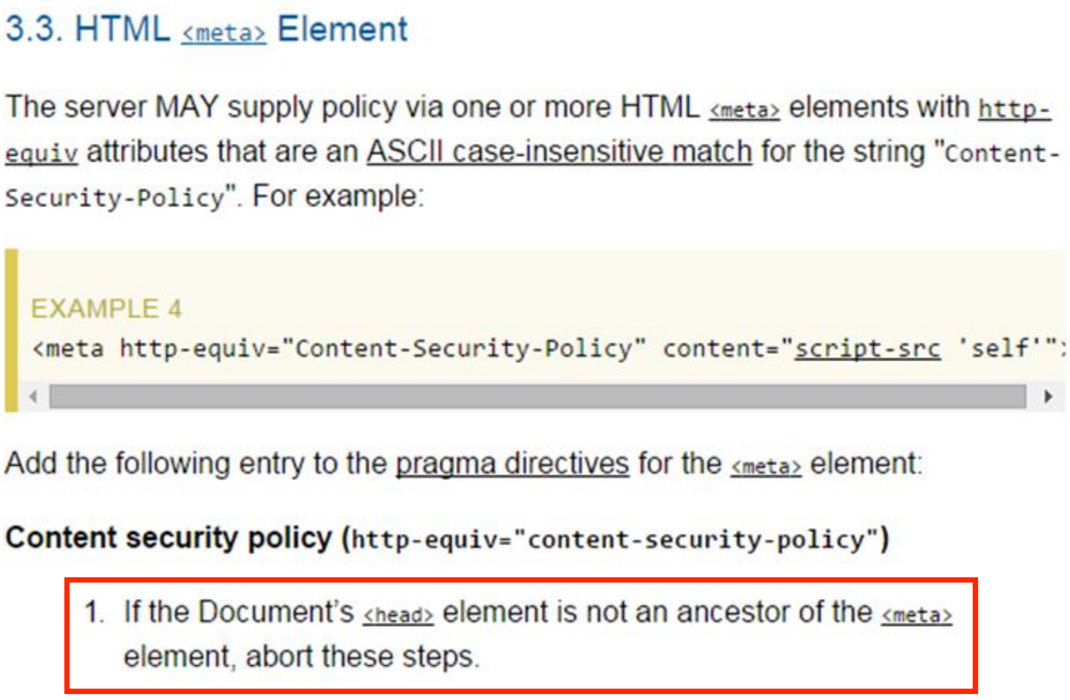
To take advantage of CSP, a web application opts into using CSP by supplying a `Content-Security-Policy` HTTP header. Such policies apply to the current resource representation only. To supply a policy for an entire site, the server needs to supply a policy with each resource representation.

Source: http://www.w3.org/TR/CSP/

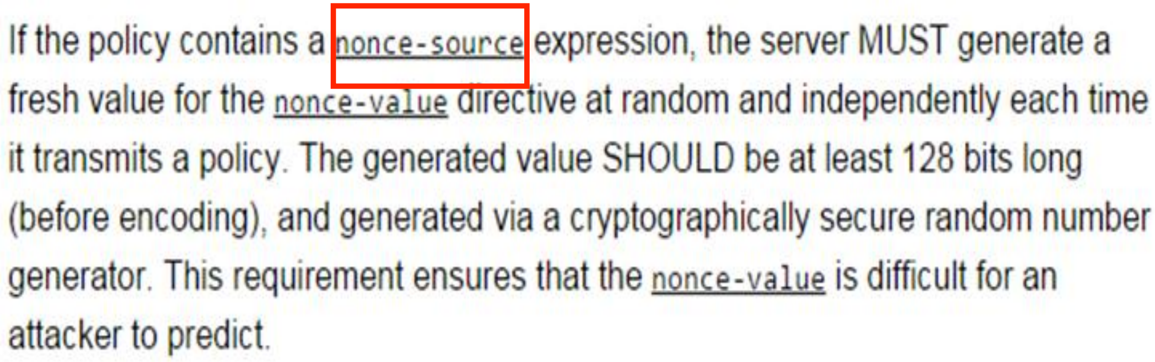
US 8104092 – Claim 10 vs. W3C standards

Claim	W3C
10. **A method for document integrity,** comprising: detecting an initiator insertion point in an electronic document; generating a key; associating the key with an initiator at the initiator insertion point; associating the key with a terminator at a terminator insertion point, wherein the terminator corresponds to the initiator; and saving the document.	As shown in the snapshot below, Content Security Policy prevents cross-site scripting attacks and other related vulnerabilities. The policy is to deliver security benefits to developers to protect their scripts. Thus, Content Security Policy provides document integrity.  1. Introduction *This section is not normative.* This document defines Content Security Policy, a mechanism web applications can use to mitigate a broad class of content injection vulnerabilities, such as cross-site scripting (XSS). Content Security Policy is a declarative policy that lets the authors (or server administrators) of a web application inform the client about the sources from which the application expects to load resources. To mitigate XSS attacks, for example, a web application can declare that it only expects to load script from specific, trusted sources. This declaration allows the client to detect and block malicious scripts injected into the application by an attacker. Source: http://www.w3.org/TR/CSP/

US 8104092 – Claim 10 vs. W3C standards

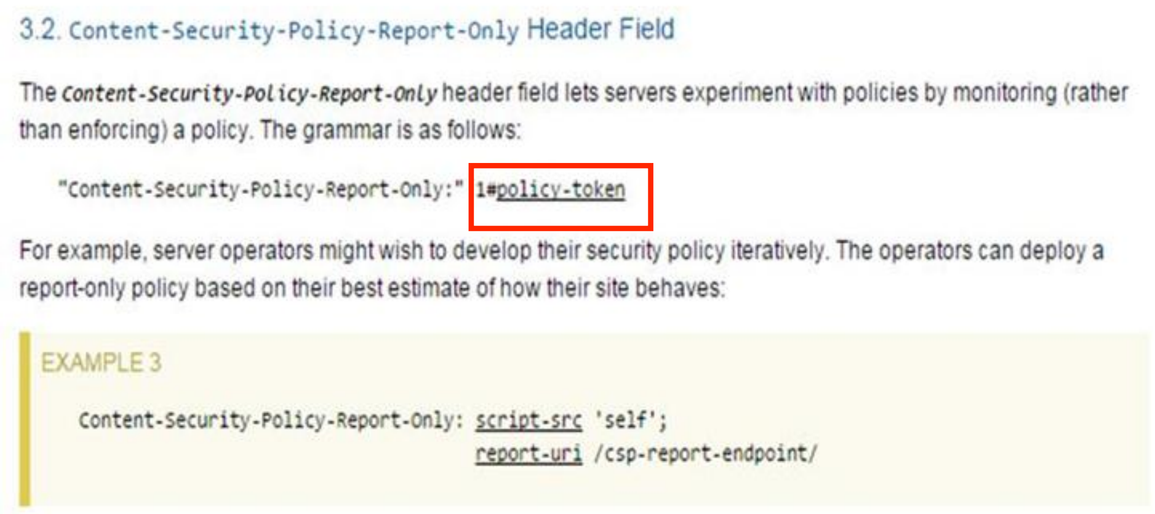
Claim	W3C
10. **A method for document integrity,** comprising: detecting an initiator insertion point in an electronic document; generating a key; associating the key with an initiator at the initiator insertion point; associating the key with a terminator at a terminator insertion point, wherein the terminator corresponds to the initiator; and saving the document.	As shown in the snapshots below, Content Security Policy is used for document integrity.  ### 3.3. HTML `<meta>` Element The server MAY supply policy via one or more HTML `<meta>` elements with `http-equiv` attributes that are an ASCII case-insensitive match for the string "`Content-Security-Policy`". For example: EXAMPLE 4 `<meta http-equiv="Content-Security-Policy" content="script-src 'self'"` Add the following entry to the pragma directives for the `<meta>` element: **Content security policy** (`http-equiv="content-security-policy"`) 1. If the Document's `<head>` element is not an ancestor of the `<meta>` element, abort these steps. Source: http://www.w3.org/TR/CSP/

US 8104092 – Claim 10 vs. W3C standards

Claim	W3C
10. A method for document integrity, comprising: **detecting an initiator insertion point in an electronic document;** generating a key; associating the key with an initiator at the initiator insertion point; associating the key with a terminator at a terminator insertion point, wherein the terminator corresponds to the initiator; and saving the document.	As shown in the snapshot below, nonce - source expression is the initiator. It is included in the Content Security Policy header which is the insertion point.  If the policy contains a nonce-source expression, the server MUST generate a fresh value for the nonce-value directive at random and independently each time it transmits a policy. The generated value SHOULD be at least 128 bits long (before encoding), and generated via a cryptographically secure random number generator. This requirement ensures that the nonce-value is difficult for an attacker to predict. Source: http://www.w3.org/TR/CSP/

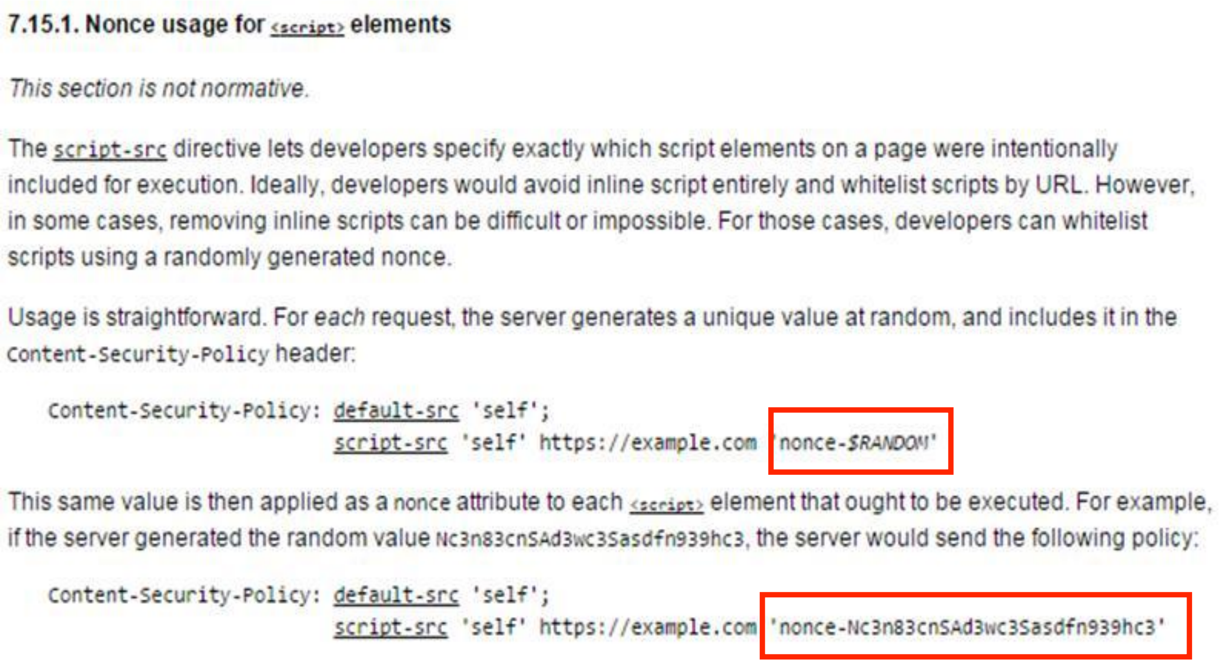
US 8104092 – Claim 10 vs. W3C standards

Claim	W3C

10. A method for document integrity, comprising: **detecting an initiator insertion point in an electronic document;** generating a key; associating the key with an initiator at the initiator insertion point; associating the key with a terminator at a terminator insertion point, wherein the terminator corresponds to the initiator; and saving the document.	As shown in the snapshot below, Content Security policy is the insertion point. Nonce initiator is included in place of the ‘policy token’. A HTML file is an electronic document and Content Security policy is set in order to maintain the integrity of the elements of HTML file.
	
	3.2. `Content-Security-Policy-Report-Only` Header Field
	The `Content-Security-Policy-Report-Only` header field lets servers experiment with policies by monitoring (rather than enforcing) a policy. The grammar is as follows:
	`"Content-Security-Policy-Report-Only:" 1#policy-token`
	For example, server operators might wish to develop their security policy iteratively. The operators can deploy a report-only policy based on their best estimate of how their site behaves:
	EXAMPLE 3
	`Content-Security-Policy-Report-Only: script-src 'self';`
	`report-uri /csp-report-endpoint/`
	Source: http://www.w3.org/TR/CSP/

US 8104092 – Claim 10 vs. W3C standards

Claim	W3C

10. A method for document integrity, comprising: **detecting an initiator insertion point in an electronic document;** generating a key; associating the key with an initiator at the initiator insertion point; associating the key with a terminator at a terminator insertion point, wherein the terminator corresponds to the initiator; and saving the document.	As shown in the snapshot below, 'nonce-$RANDOM' is included in the Content-Security-Policy header (insertion point). 'Nonce-$RANDOM' is the initiator. An example has been highlighted below in the screenshot. A HTML file is an electronic document and Content Security policy is set in order to maintain the integrity of the elements of HTML file.
	
	7.15.1. Nonce usage for `<script>` elements
	This section is not normative.
	The `script-src` directive lets developers specify exactly which script elements on a page were intentionally included for execution. Ideally, developers would avoid inline script entirely and whitelist scripts by URL. However, in some cases, removing inline scripts can be difficult or impossible. For those cases, developers can whitelist scripts using a randomly generated nonce.
	Usage is straightforward. For *each* request, the server generates a unique value at random, and includes it in the `Content-Security-Policy` header:
	`Content-Security-Policy: default-src 'self';`
	`script-src 'self' https://example.com 'nonce-$RANDOM'`
	This same value is then applied as a `nonce` attribute to each `<script>` element that ought to be executed. For example, if the server generated the random value `Nc3n83cnSAd3wc3Sasdfn939hc3`, the server would send the following policy:
	`Content-Security-Policy: default-src 'self';`
	`script-src 'self' https://example.com 'nonce-Nc3n83cnSAd3wc3Sasdfn939hc3'`
	Source: http://www.w3.org/TR/CSP/

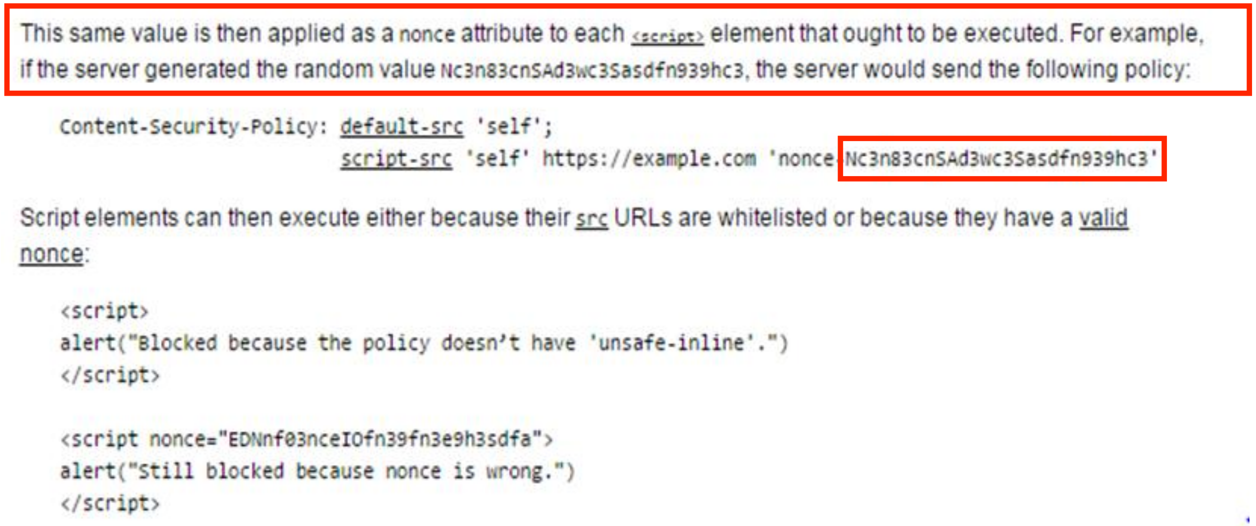
US 8104092 – Claim 10 vs. W3C standards

Claim

10. A method for document integrity, comprising: detecting an initiator insertion point in an electronic document; **generating a key;** associating the key with an initiator at the initiator insertion point; associating the key with a terminator at a terminator insertion point, wherein the terminator corresponds to the initiator; and saving the document.

W3C

As shown in the snapshot below, the server generates a random value for the nonce-value expression. This random value is the key (Nc3n83cnSAd3wc3Sasdfn939hc3). This key is applied to each element of the script as the nonce attribute.



This same value is then applied as a nonce attribute to each <script> element that ought to be executed. For example, if the server generated the random value Nc3n83cnSAd3wc3Sasdfn939hc3, the server would send the following policy:

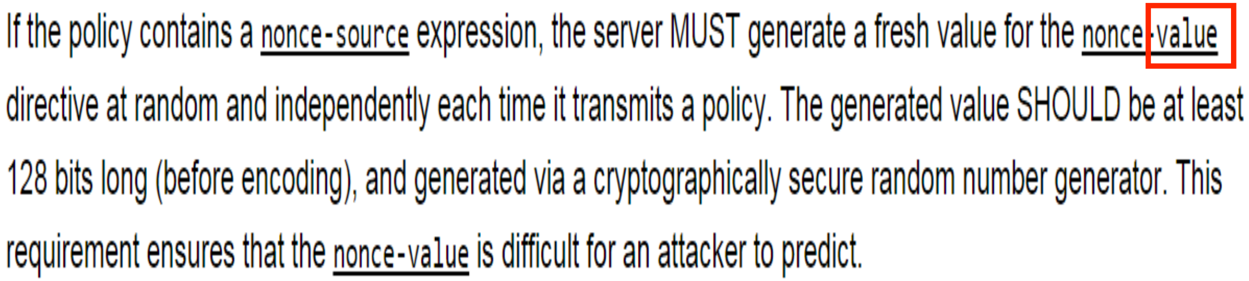
```
Content-Security-Policy: default-src 'self';
                         script-src 'self' https://example.com 'nonce-Nc3n83cnSAd3wc3Sasdfn939hc3'
```

Script elements can then execute either because their src URLs are whitelisted or because they have a valid nonce:

```
<script>
alert("Blocked because the policy doesn’t have 'unsafe-inline'.")
</script>

<script nonce="EDNnf03nceIOfn39fn3e9h3sdfa">
alert("Still blocked because nonce is wrong.")
</script>
```

Source: http://www.w3.org/TR/CSP/

US 8104092 – Claim 10 vs. W3C standards

Claim	W3C
10. A method for document integrity, comprising: detecting an initiator insertion point in an electronic document; **generating a key;** associating the key with an initiator at the initiator insertion point; associating the key with a terminator at a terminator insertion point, wherein the terminator corresponds to the initiator; and saving the document.	As shown in the snapshot below, the highlighted random value associated with the nonce (initiator) is the generated key.  If the policy contains a nonce-source expression, the server MUST generate a fresh value for the nonce-value directive at random and independently each time it transmits a policy. The generated value SHOULD be at least 128 bits long (before encoding), and generated via a cryptographically secure random number generator. This requirement ensures that the nonce-value is difficult for an attacker to predict. Source: http://www.w3.org/TR/CSP/

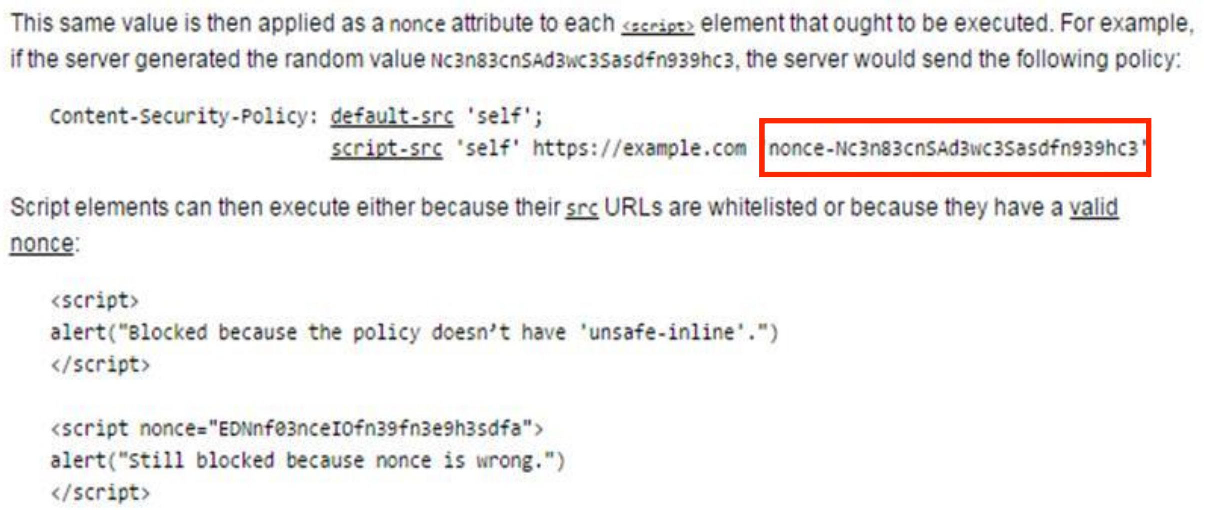
US 8104092 – Claim 10 vs. W3C standards

Claim

10. A method for document integrity, comprising: detecting an initiator insertion point in an electronic document; generating a key; **associating the key with an initiator at the initiator insertion point;** associating the key with a terminator at a terminator insertion point, wherein the terminator corresponds to the initiator; and saving the document.

W3C

As shown in the snapshot below, Nc3n83cnSAd3wc3Sasdfn939hc3 (key) is associated with nonce (initiator) in the Content Security Policy (initiator insertion point).



This same value is then applied as a nonce attribute to each `<script>` element that ought to be executed. For example, if the server generated the random value Nc3n83cnSAd3wc3Sasdfn939hc3, the server would send the following policy:

```
Content-Security-Policy: default-src 'self';
                         script-src 'self' https://example.com 'nonce-Nc3n83cnSAd3wc3Sasdfn939hc3'
```

Script elements can then execute either because their src URLs are whitelisted or because they have a valid nonce:

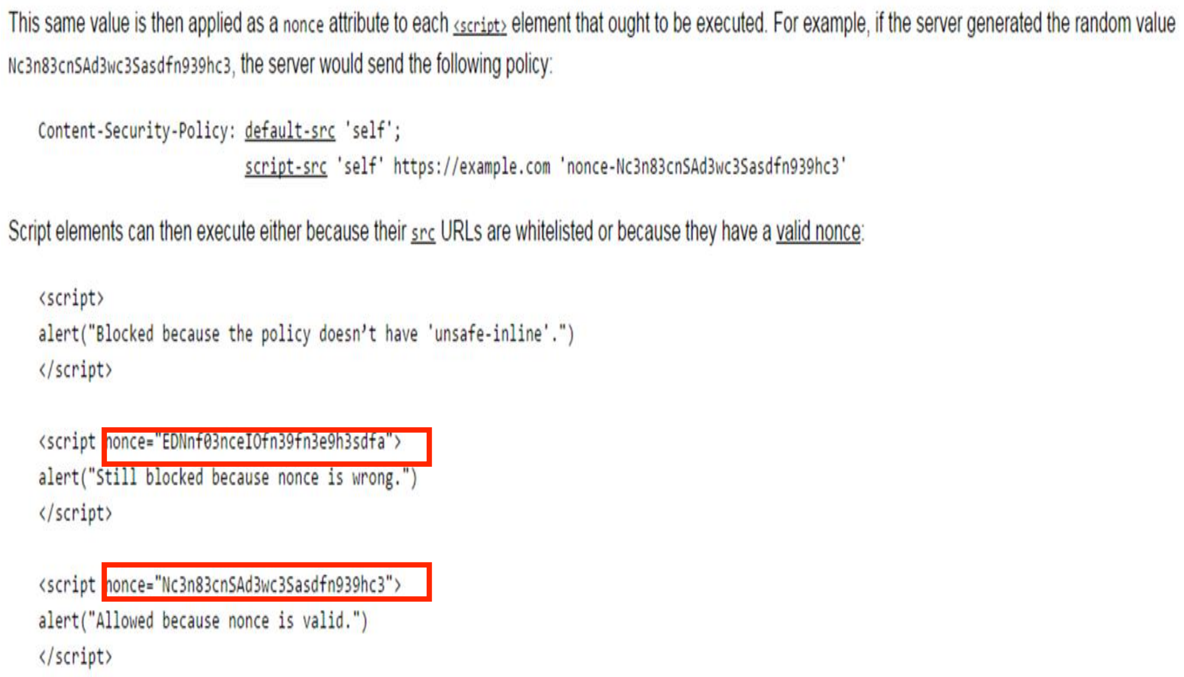
```
<script>
alert("Blocked because the policy doesn't have 'unsafe-inline'.")
</script>

<script nonce="EDNnf03nceIOfn39fn3e9h3sdfa">
alert("Still blocked because nonce is wrong.")
</script>
```

Source: http://www.w3.org/TR/CSP/

© ICEBERG Capital Partners Limited 2015. Confidential

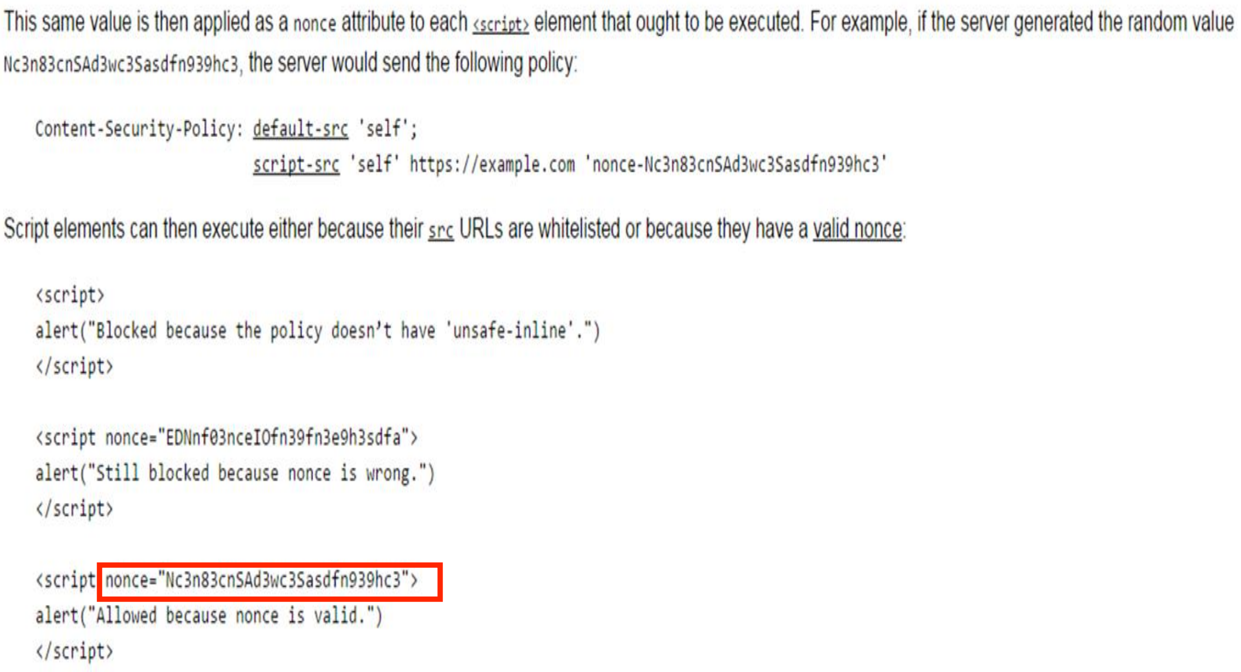
US 8104092 – Claim 10 vs. W3C standards

Claim	W3C

10. A method for document integrity, comprising: detecting an initiator insertion point in an electronic document; generating a key; associating the key with an initiator at the initiator insertion point; **associating the key with a terminator at a terminator insertion point,** wherein the terminator corresponds to the initiator; and saving the document.	As shown in the snapshot below, there are two termination insertion points. First one is when the script is invalid and here the terminator (nonce) is linked with key which does not match with the key value at the insertion point.. The second one is when the script is valid and here the terminator is associated with key which matches the key value of the initiator.
	
	This same value is then applied as a `nonce` attribute to each `<script>` element that ought to be executed. For example, if the server generated the random value `Nc3n83cnSAd3wc3Sasdfn939hc3`, the server would send the following policy:
	`Content-Security-Policy: default-src 'self';`
	`script-src 'self' https://example.com 'nonce-Nc3n83cnSAd3wc3Sasdfn939hc3'`
	Script elements can then execute either because their `src` URLs are whitelisted or because they have a valid nonce:
	`<script>`
	`alert("Blocked because the policy doesn’t have 'unsafe-inline'.")`
	`</script>`
	`<script nonce="EDNnf03nceIOfn39fn3e9h3sdfa">`
	`alert("Still blocked because nonce is wrong.")`
	`</script>`
	`<script nonce="Nc3n83cnSAd3wc3Sasdfn939hc3">`
	`alert("Allowed because nonce is valid.")`
	`</script>`
	Source: http://www.w3.org/TR/CSP/

US 8104092 – Claim 10 vs. W3C standards

Claim	W3C

10. A method for document integrity, comprising: detecting an initiator insertion point in an electronic document; generating a key; associating the key with an initiator at the initiator insertion point; **associating the key with a terminator at a terminator insertion point,** wherein the terminator corresponds to the initiator; and saving the document.	As shown in the snapshot below, script elements are sent by the server with a nonce value attached to it. This nonce value can be construed to be the terminator. The nonce is associated with the key (Nc3n83cnSAd3wc3Sasdfn939hc3) when the document is valid. This terminator key value matches the key value of the initiator.
	
	This same value is then applied as a `nonce` attribute to each `<script>` element that ought to be executed. For example, if the server generated the random value `Nc3n83cnSAd3wc3Sasdfn939hc3`, the server would send the following policy:
	`Content-Security-Policy: default-src 'self';`
	`script-src 'self' https://example.com 'nonce-Nc3n83cnSAd3wc3Sasdfn939hc3'`
	Script elements can then execute either because their `src` URLs are whitelisted or because they have a valid nonce:
	`<script>`
	`alert("Blocked because the policy doesn't have 'unsafe-inline'.")`
	`</script>`
	`<script nonce="EDNnf03nceIOfn39fn3e9h3sdfa">`
	`alert("Still blocked because nonce is wrong.")`
	`</script>`
	`<script nonce="Nc3n83cnSAd3wc3Sasdfn939hc3">`
	`alert("Allowed because nonce is valid.")`
	`</script>`
	Source: http://www.w3.org/TR/CSP/

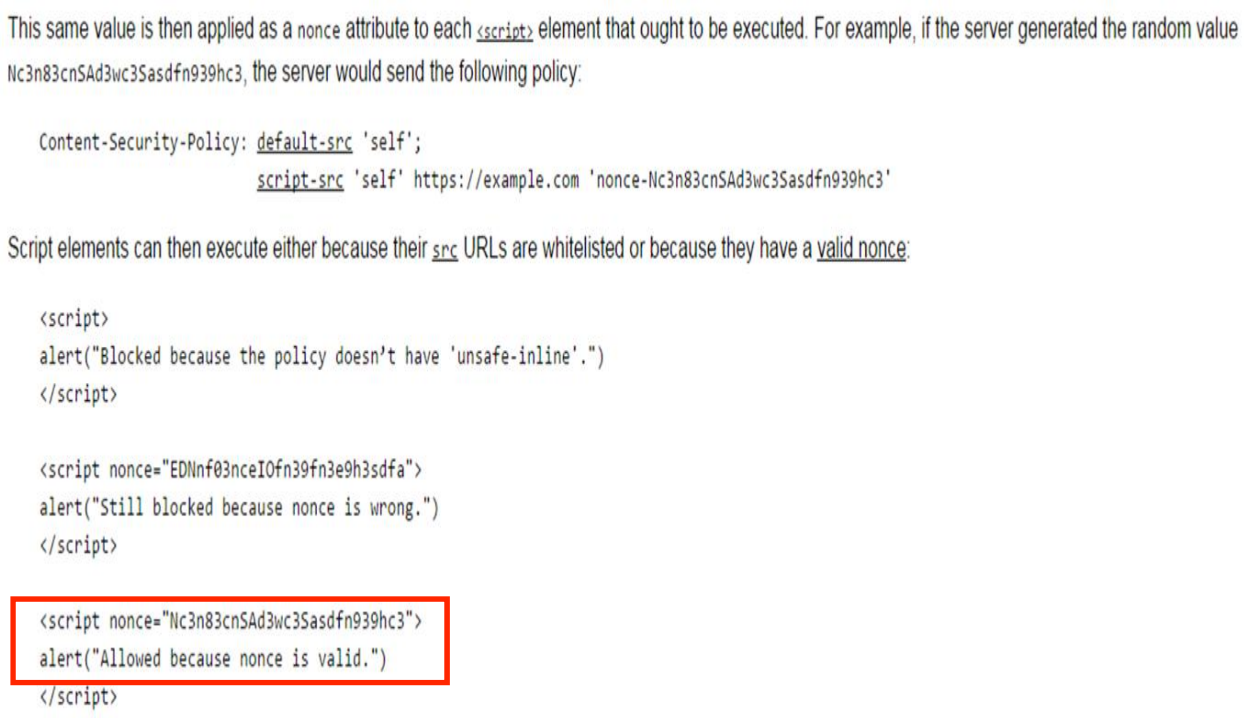
US 8104092 – Claim 10 vs. W3C standards

Claim

10. A method for document integrity, comprising: detecting an initiator insertion point in an electronic document; generating a key; associating the key with an initiator at the initiator insertion point; associating the key with a terminator at a terminator insertion point, wherein **the terminator corresponds to the initiator**; and saving the document.

W3C

As shown in the snapshot below, the value of nonce(terminator) in the termination section is checked if with the value of nonce(initiator) in the Content Security Policy header. Thus, the terminator corresponds with the initiator.



This same value is then applied as a `nonce` attribute to each `<script>` element that ought to be executed. For example, if the server generated the random value `Nc3n83cnSAd3wc3Sasdfn939hc3`, the server would send the following policy:

```
Content-Security-Policy: default-src 'self';
                         script-src 'self' https://example.com 'nonce-Nc3n83cnSAd3wc3Sasdfn939hc3'
```

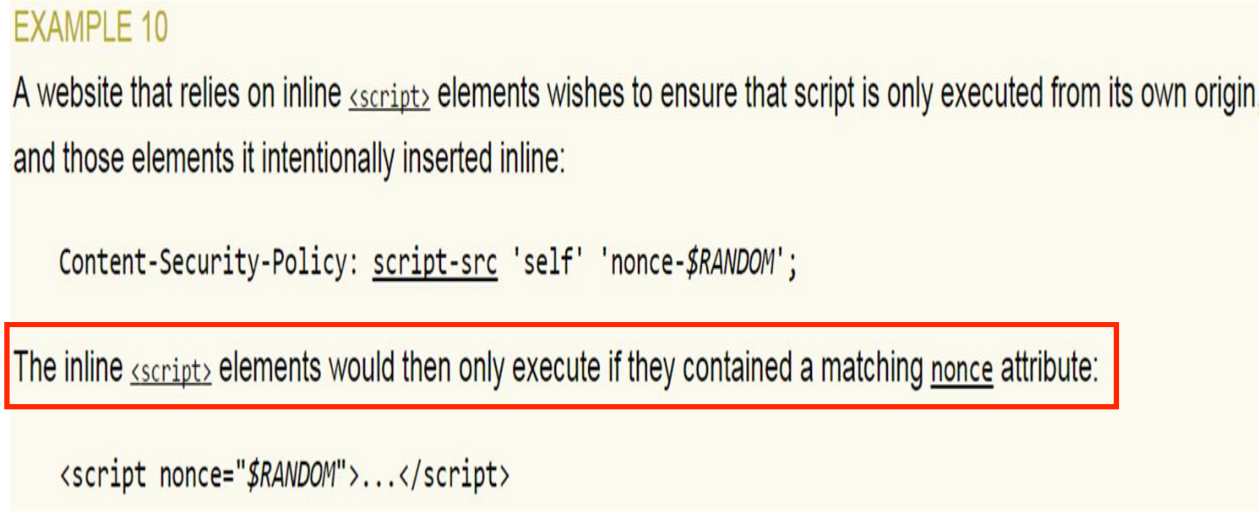
Script elements can then execute either because their `src` URLs are whitelisted or because they have a valid nonce:

```
<script>
alert("Blocked because the policy doesn't have 'unsafe-inline'.")
</script>

<script nonce="EDNnf03nceIOfn39fn3e9h3sdfa">
alert("Still blocked because nonce is wrong.")
</script>

<script nonce="Nc3n83cnSAd3wc3Sasdfn939hc3">
alert("Allowed because nonce is valid.")
</script>
```

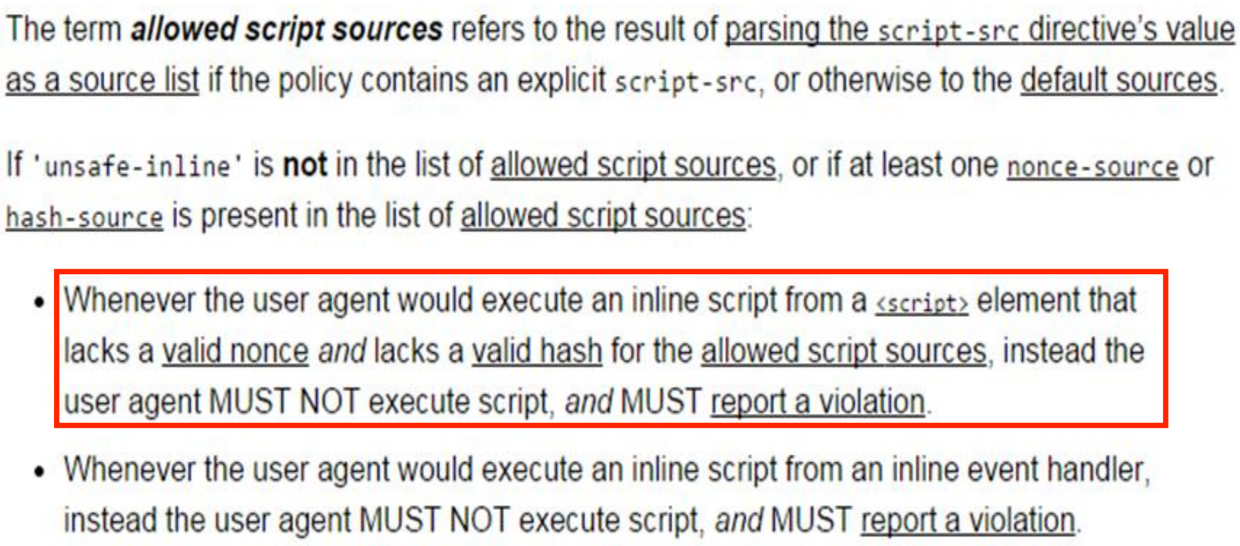
Source: http://www.w3.org/TR/CSP/

US 8104092 – Claim 10 vs. W3C standards

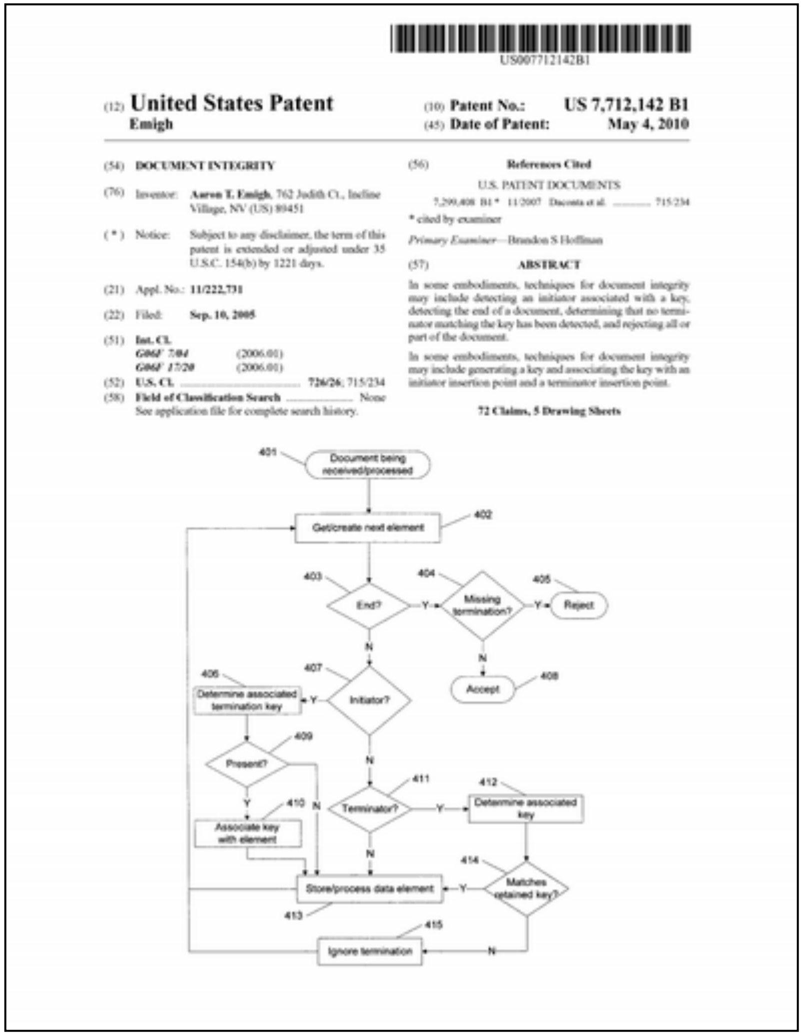
Claim	W3C

10. A method for document integrity, comprising: detecting an initiator insertion point in an electronic document; generating a key; associating the key with an initiator at the initiator insertion point; associating the key with a terminator at a terminator insertion point, wherein the terminator corresponds to the initiator; and **saving the document**.	As shown in the snapshot below, if the key value of the initiator and terminator matches, the script (document) is executed (saved). The execution takes place after the script is saved in the browser software.
	
	EXAMPLE 10
	A website that relies on inline `<script>` elements wishes to ensure that script is only executed from its own origin, and those elements it intentionally inserted inline:
	`Content-Security-Policy: script-src 'self' 'nonce-$RANDOM';`
	The inline `<script>` elements would then only execute if they contained a matching nonce attribute:
	`<script nonce="$RANDOM">...</script>`
	Source: http://www.w3.org/TR/CSP/

US 8104092 – Claim 10 vs. W3C standards

Claim	W3C
10. A method for document integrity, comprising: detecting an initiator insertion point in an electronic document; generating a key; associating the key with an initiator at the initiator insertion point; associating the key with a terminator at a terminator insertion point, wherein the terminator corresponds to the initiator; and **saving the document**.	As shown in the snapshot below, if the key value of the initiator and terminator matches, the script (document) is executed (saved). The execution takes place after the script is saved in the browser software.  The term ***allowed script sources*** refers to the result of parsing the `script-src` directive's value as a source list if the policy contains an explicit `script-src`, or otherwise to the default sources. If `'unsafe-inline'` is **not** in the list of allowed script sources, or if at least one `nonce-source` or `hash-source` is present in the list of allowed script sources: - Whenever the user agent would execute an inline script from a `<script>` element that lacks a valid nonce *and* lacks a valid hash for the allowed script sources, instead the user agent MUST NOT execute script, *and* MUST report a violation. - Whenever the user agent would execute an inline script from an inline event handler, instead the user agent MUST NOT execute script, *and* MUST report a violation. Source: http://www.w3.org/TR/CSP/

Family 1: US 7712142 – Bibliographic information



US007712142B1

(12) **United States Patent**
Emigh

(10) **Patent No.:** **US 7,712,142 B1**
(45) **Date of Patent:** **May 4, 2010**

(54) **DOCUMENT INTEGRITY**

(76) Inventor: **Aaron T. Emigh**, 762 Judith Ct., Incline Village, NV (US) 89451

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 1221 days.

(21) Appl. No.: **11/222,731**

(22) Filed: **Sep. 10, 2005**

(51) **Int. Cl.**
G06F 7/04 (2006.01)
G06F 17/20 (2006.01)

(52) **U.S. Cl.** **726/26**; 715/234

(58) **Field of Classification Search** .... None
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

7,299,408 B1 * 11/2007 Daconta et al. 715/234

* cited by examiner

Primary Examiner—Brandon S Hoffman

(57) **ABSTRACT**

In some embodiments, techniques for document integrity may include detecting an initiator associated with a key, detecting the end of a document, determining that no terminator matching the key has been detected, and rejecting all or part of the document.

In some embodiments, techniques for document integrity may include generating a key and associating the key with an initiator insertion point and a terminator insertion point.

72 Claims, 5 Drawing Sheets

Patent of Interest:

US7712142
(Priority date: Sept 10, 2005)

Document Integrity

Exemplary Market Applications:

The patented technology finds applications in document security.

US 7712142 – Claim 25

25. A method for document integrity, comprising:

detecting an initiator insertion point in an electronic document;

generating a key;

associating the key with an initiator at the initiator insertion point;

associating the key with a terminator at a termination insertion point, wherein the terminator corresponds to the initiator; and

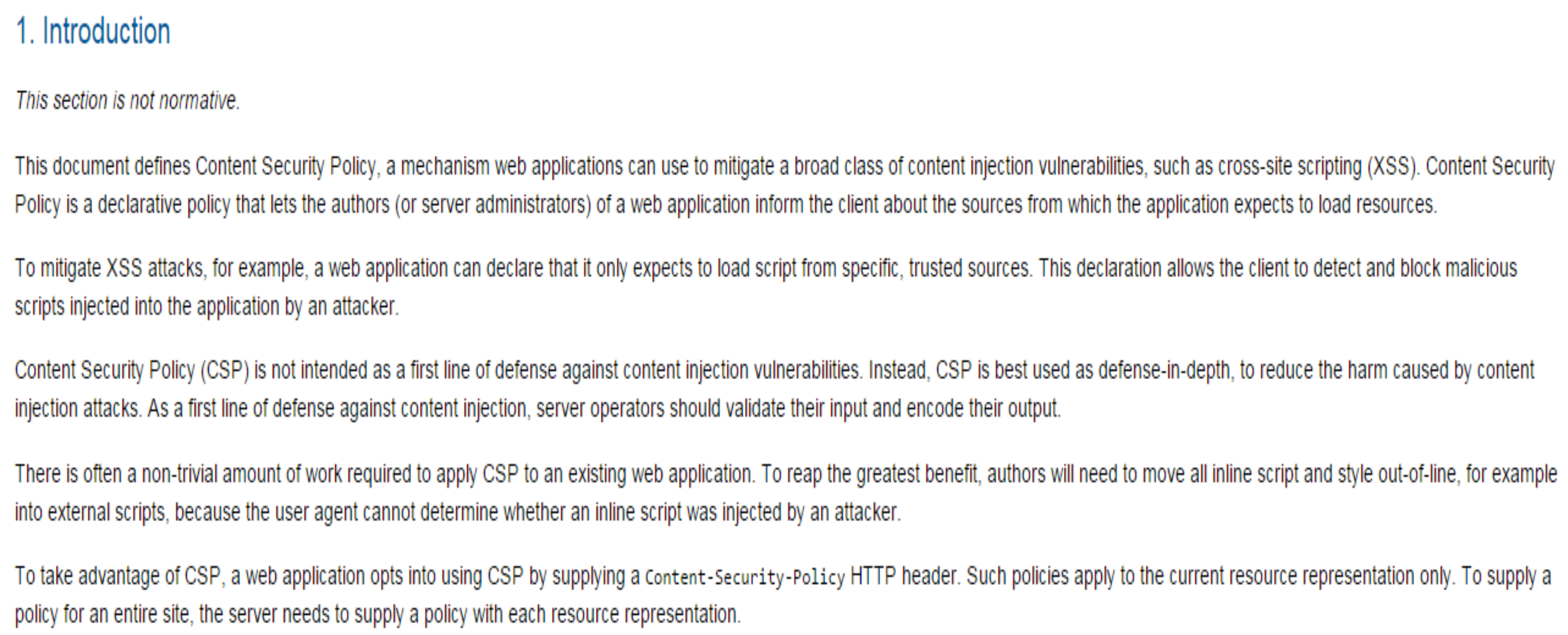
electronically transmitting the document.

US 7712142 – EoU Summary

W3C	
Key claim(s)	1, 13, 25, 37, 49, and 61 (Independent Claims)
Mapped product	Chart has been made with respect to W3C. http://www.w3.org/TR/CSP/
Source	Information related to product is available at: http://www.w3.org/TR/CSP/
Product launch date	Content Security Policy specification was launched in 2012
Details of standard	The World Wide Consortium (W3C) is an international standard organization for World Wide Web. The Candidate Recommendation of this organization includes a policy called Content Security Policy. This policy is for computer security as it prevents attacks like cross-site scripting.

US 7712142 – W3C: Overview

The World Wide Consortium (W3C) is an international standard organization for World Wide Web. The Candidate Recommendation of this organization includes a policy called Content Security Policy. This policy is for computer security as it prevents attacks like cross-site scripting.



1. Introduction

This section is not normative.

This document defines Content Security Policy, a mechanism web applications can use to mitigate a broad class of content injection vulnerabilities, such as cross-site scripting (XSS). Content Security Policy is a declarative policy that lets the authors (or server administrators) of a web application inform the client about the sources from which the application expects to load resources.

To mitigate XSS attacks, for example, a web application can declare that it only expects to load script from specific, trusted sources. This declaration allows the client to detect and block malicious scripts injected into the application by an attacker.

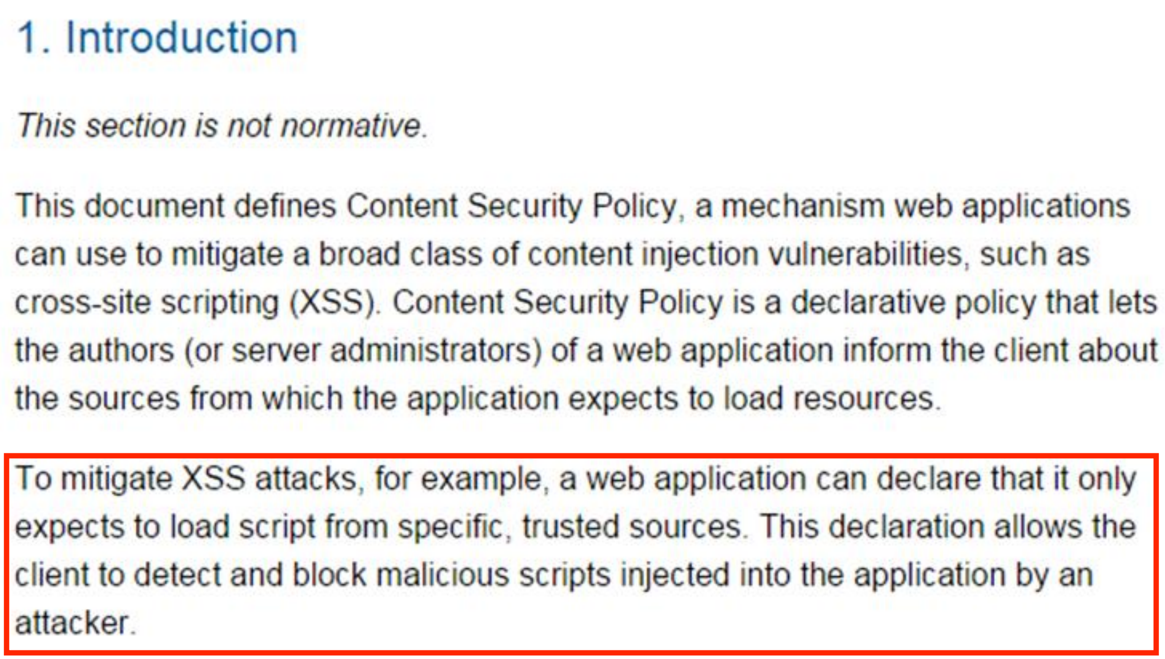
Content Security Policy (CSP) is not intended as a first line of defense against content injection vulnerabilities. Instead, CSP is best used as defense-in-depth, to reduce the harm caused by content injection attacks. As a first line of defense against content injection, server operators should validate their input and encode their output.

There is often a non-trivial amount of work required to apply CSP to an existing web application. To reap the greatest benefit, authors will need to move all inline script and style out-of-line, for example into external scripts, because the user agent cannot determine whether an inline script was injected by an attacker.

To take advantage of CSP, a web application opts into using CSP by supplying a `Content-Security-Policy` HTTP header. Such policies apply to the current resource representation only. To supply a policy for an entire site, the server needs to supply a policy with each resource representation.

Source: http://www.w3.org/TR/CSP/

US 7712142 – Claim 25 vs. W3C standards

Claim	W3C
25. **A method for document integrity, comprising:** detecting an initiator insertion point in an electronic document; generating a key; associating the key with an initiator at the initiator insertion point; associating the key with a terminator at a termination insertion point, wherein the terminator corresponds to the initiator; and electronically transmitting the document.	As shown in the snapshot below, Content Security Policy prevents cross-site scripting attacks and other related vulnerabilities. The policy is to deliver security benefits to developers to protect their scripts. Thus, Content Security Policy provides document integrity.  1. Introduction *This section is not normative.* This document defines Content Security Policy, a mechanism web applications can use to mitigate a broad class of content injection vulnerabilities, such as cross-site scripting (XSS). Content Security Policy is a declarative policy that lets the authors (or server administrators) of a web application inform the client about the sources from which the application expects to load resources. To mitigate XSS attacks, for example, a web application can declare that it only expects to load script from specific, trusted sources. This declaration allows the client to detect and block malicious scripts injected into the application by an attacker. Source: http://www.w3.org/TR/CSP/

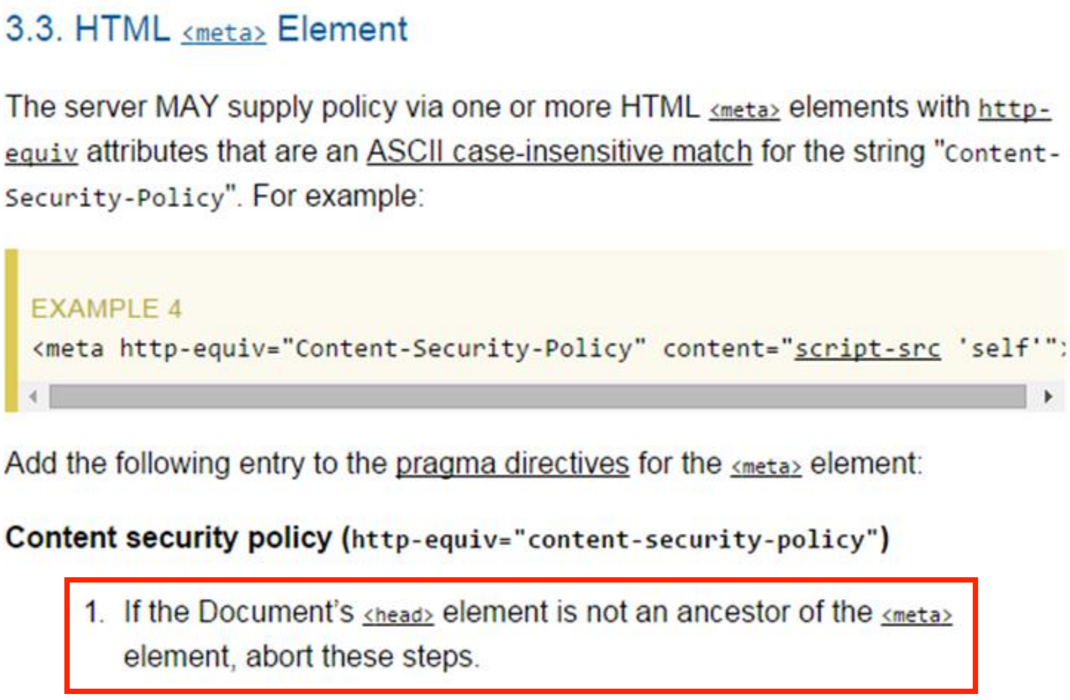
US 7712142 – Claim 25 vs. W3C standards

Claim

25. **A method for document integrity, comprising:** detecting an initiator insertion point in an electronic document; generating a key; associating the key with an initiator at the initiator insertion point; associating the key with a terminator at a termination insertion point, wherein the terminator corresponds to the initiator; and electronically transmitting the document.

W3C

As shown in the snapshot below, Content Security Policy is used for document integrity.



3.3. HTML `<meta>` Element

The server MAY supply policy via one or more HTML `<meta>` elements with `http-equiv` attributes that are an ASCII case-insensitive match for the string "`Content-Security-Policy`". For example:

EXAMPLE 4

```
<meta http-equiv="Content-Security-Policy" content="script-src 'self'";
```

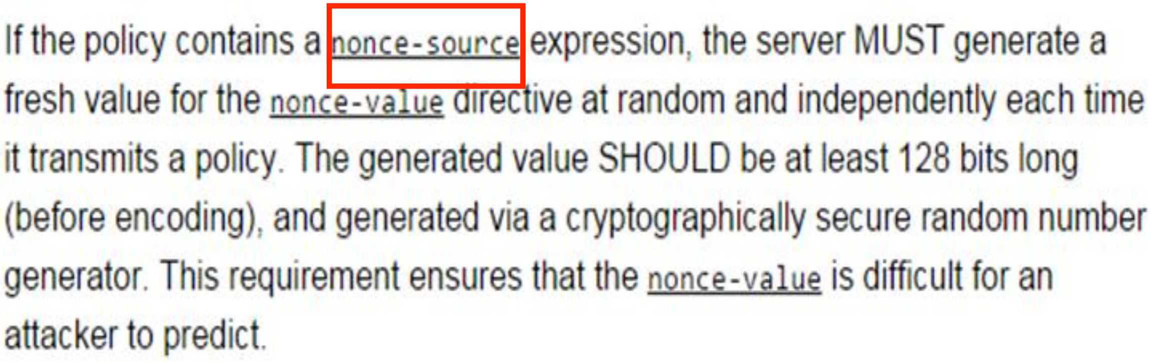
Add the following entry to the pragma directives for the `<meta>` element:

Content security policy (`http-equiv="content-security-policy"`)

1. If the Document's `<head>` element is not an ancestor of the `<meta>` element, abort these steps.

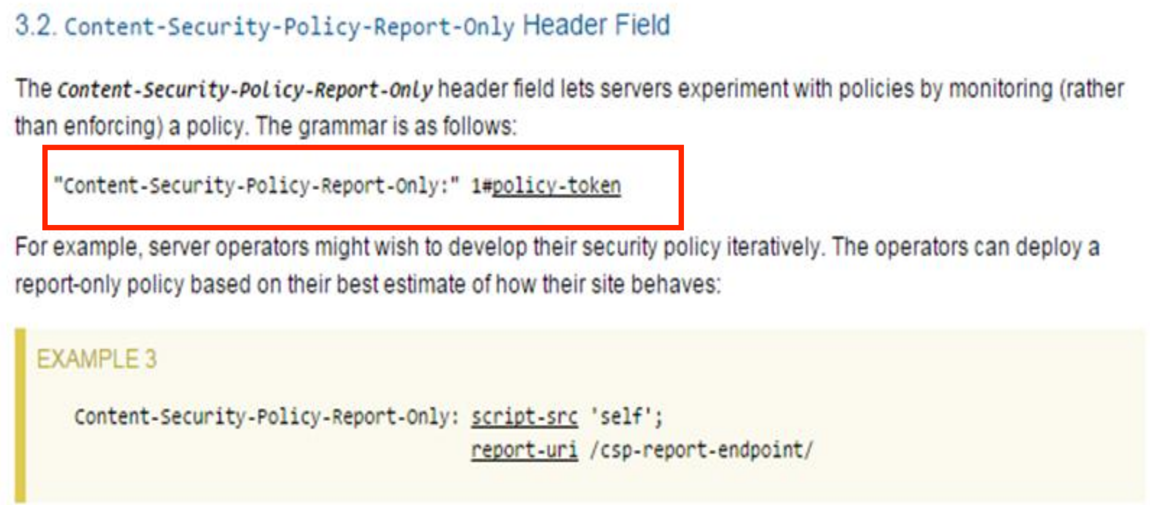
Source: http://www.w3.org/TR/CSP/

US 7712142 – Claim 25 vs. W3C standards

Claim	W3C
25. A method for document integrity, comprising: **detecting an initiator insertion point in an electronic document;** generating a key; associating the key with an initiator at the initiator insertion point; associating the key with a terminator at a termination insertion point, wherein the terminator corresponds to the initiator; and electronically transmitting the document.	As shown in the snapshot below, nonce - source expression is the initiator. It is included in the Content Security Policy header which is the insertion point.  If the policy contains a nonce-source expression, the server MUST generate a fresh value for the nonce-value directive at random and independently each time it transmits a policy. The generated value SHOULD be at least 128 bits long (before encoding), and generated via a cryptographically secure random number generator. This requirement ensures that the nonce-value is difficult for an attacker to predict. Source: http://www.w3.org/TR/CSP/

US 7712142 – Claim 25 vs. W3C standards

Claim	W3C

25. A method for document integrity, comprising: **detecting an initiator insertion point in an electronic document;** generating a key; associating the key with an initiator at the initiator insertion point; associating the key with a terminator at a termination insertion point, wherein the terminator corresponds to the initiator; and electronically transmitting the document.	As shown in the snapshot below, Content Security policy header is the insertion point. Nonce initiator is included in place of the 'policy token'. A HTML file is an electronic document and Content Security policy is set in order to maintain the integrity of the elements of HTML file.
	
	3.2. `Content-Security-Policy-Report-Only` Header Field
	The *`Content-Security-Policy-Report-Only`* header field lets servers experiment with policies by monitoring (rather than enforcing) a policy. The grammar is as follows:
	`"Content-Security-Policy-Report-Only:" 1#policy-token`
	For example, server operators might wish to develop their security policy iteratively. The operators can deploy a report-only policy based on their best estimate of how their site behaves:
	EXAMPLE 3
	`Content-Security-Policy-Report-Only: script-src 'self';`
	`report-uri /csp-report-endpoint/`
	Source: http://www.w3.org/TR/CSP/

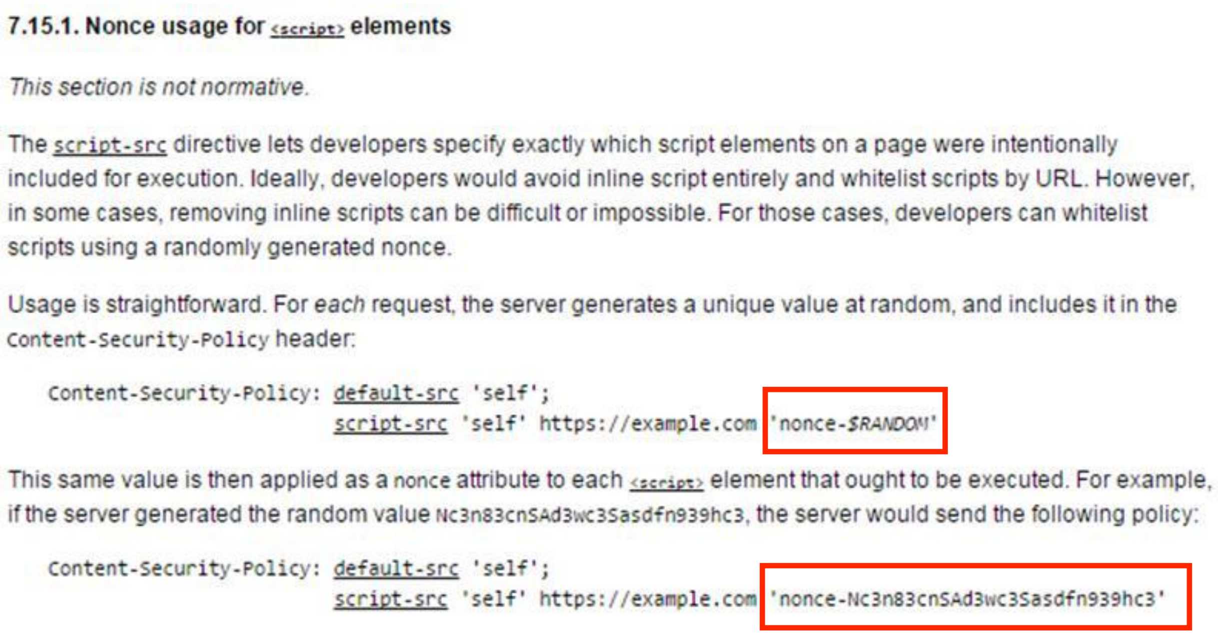
US 7712142 – Claim 25 vs. W3C standards

Claim

25. A method for document integrity, comprising: **detecting an initiator insertion point in an electronic document;** generating a key; associating the key with an initiator at the initiator insertion point; associating the key with a terminator at a termination insertion point, wherein the terminator corresponds to the initiator; and electronically transmitting the document.

W3C

As shown in the snapshot below, 'nonce-$RANDOM' is included in the Content-Security-Policy header (insertion point). 'Nonce-$RANDOM' is the initiator. An example has been highlighted below in the screenshot. A HTML file is an electronic document and Content Security policy is set in order to maintain the integrity of the elements of HTML file.



7.15.1. Nonce usage for `<script>` elements

This section is not normative.

The `script-src` directive lets developers specify exactly which script elements on a page were intentionally included for execution. Ideally, developers would avoid inline script entirely and whitelist scripts by URL. However, in some cases, removing inline scripts can be difficult or impossible. For those cases, developers can whitelist scripts using a randomly generated nonce.

Usage is straightforward. For *each* request, the server generates a unique value at random, and includes it in the `Content-Security-Policy` header:

```
Content-Security-Policy: default-src 'self';
                         script-src 'self' https://example.com 'nonce-$RANDOM'
```

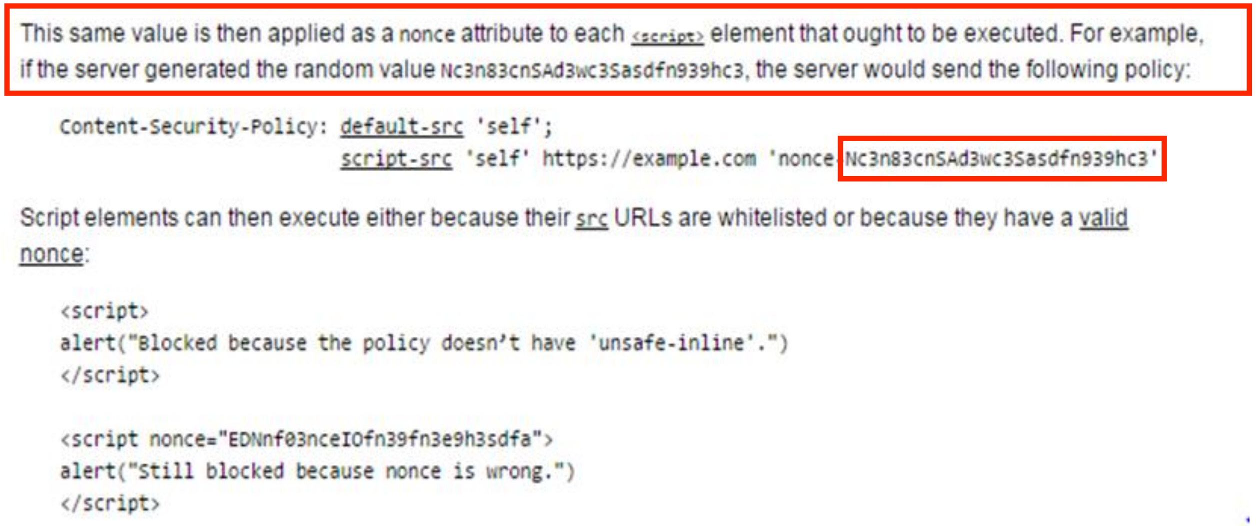
This same value is then applied as a `nonce` attribute to each `<script>` element that ought to be executed. For example, if the server generated the random value `Nc3n83cnSAd3wc3Sasdfn939hc3`, the server would send the following policy:

```
Content-Security-Policy: default-src 'self';
                         script-src 'self' https://example.com 'nonce-Nc3n83cnSAd3wc3Sasdfn939hc3'
```

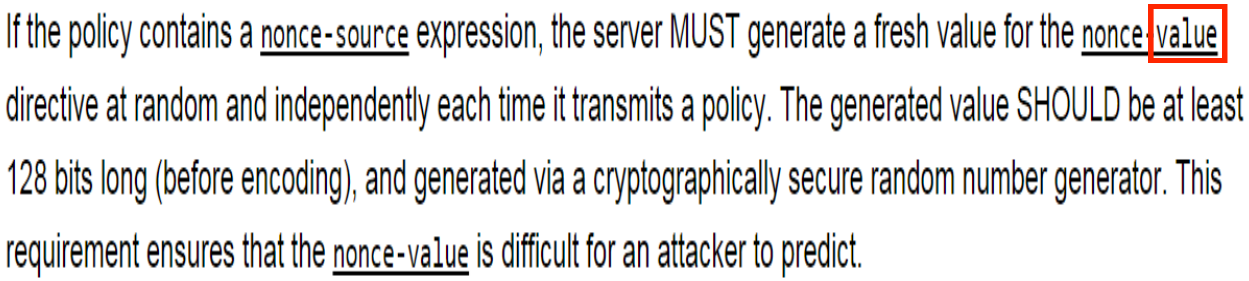
Source: http://www.w3.org/TR/CSP/

US 7712142 – Claim 25 vs. W3C standards

Claim	W3C

25. A method for document integrity, comprising: detecting an initiator insertion point in an electronic document; **generating a key;** associating the key with an initiator at the initiator insertion point; associating the key with a terminator at a termination insertion point, wherein the terminator corresponds to the initiator; and electronically transmitting the document.	As shown in the snapshot below, the server generates a random value for the nonce-value expression. This random value is the key (Nc3n83cnSAd3wc3Sasdfn939hc3). This key is applied to each element of the script as the nonce attribute.
	
	This same value is then applied as a nonce attribute to each <script> element that ought to be executed. For example, if the server generated the random value Nc3n83cnSAd3wc3Sasdfn939hc3, the server would send the following policy:
	`Content-Security-Policy: default-src 'self';`
	`                         script-src 'self' https://example.com 'nonce-Nc3n83cnSAd3wc3Sasdfn939hc3'`
	Script elements can then execute either because their src URLs are whitelisted or because they have a valid nonce:
	`<script>`
	`alert("Blocked because the policy doesn’t have 'unsafe-inline'.")`
	`</script>`
	`<script nonce="EDNnf03nceIOfn39fn3e9h3sdfa">`
	`alert("Still blocked because nonce is wrong.")`
	`</script>`
	Source: http://www.w3.org/TR/CSP/

US 7712142 – Claim 25 vs. W3C standards

Claim	W3C
25. A method for document integrity, comprising: detecting an initiator insertion point in an electronic document; **generating a key;** associating the key with an initiator at the initiator insertion point; associating the key with a terminator at a termination insertion point, wherein the terminator corresponds to the initiator; and electronically transmitting the document.	As shown in the snapshot below, the highlighted random value associated with the nonce (initiator) is the generated key.  If the policy contains a nonce-source expression, the server MUST generate a fresh value for the nonce-value directive at random and independently each time it transmits a policy. The generated value SHOULD be at least 128 bits long (before encoding), and generated via a cryptographically secure random number generator. This requirement ensures that the nonce-value is difficult for an attacker to predict. Source: http://www.w3.org/TR/CSP/

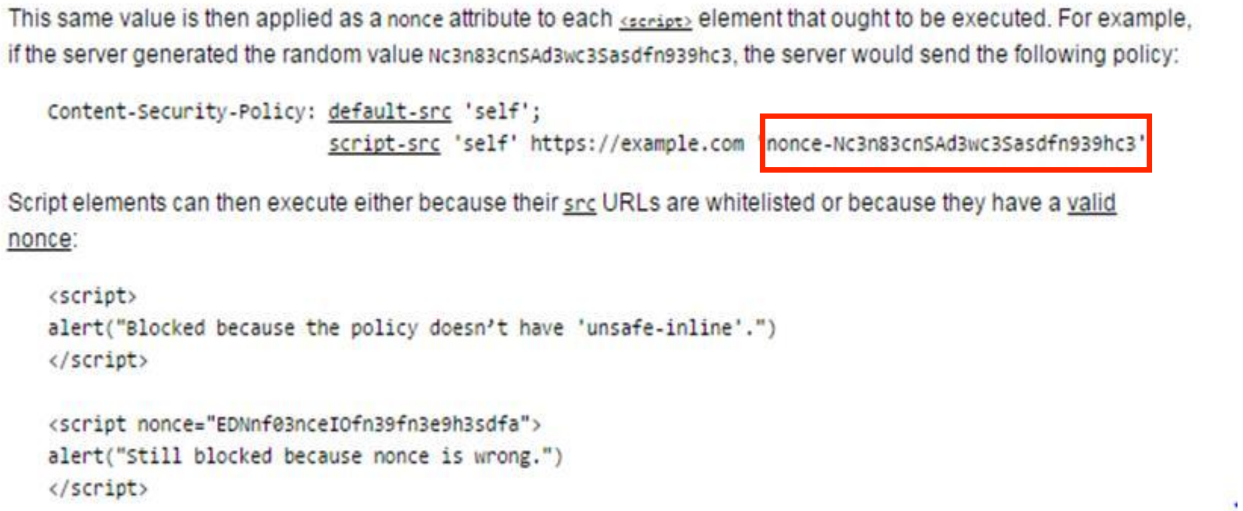
US 7712142 – Claim 25 vs. W3C standards

Claim

25. A method for document integrity, comprising: detecting an initiator insertion point in an electronic document; generating a key; **associating the key with an initiator at the initiator insertion point;** associating t he key with a terminator at a termination insertion point, wherein the terminator corresponds to the initiator; and electronically transmitting the document.

W3C

As shown in the snapshot below, key (Nc3n83cnSAd3wc3Sasdfn939hc3) is associated with nonce (initiator) in the Content Security Policy (initiator insertion point).



This same value is then applied as a nonce attribute to each <script> element that ought to be executed. For example, if the server generated the random value Nc3n83cnSAd3wc3Sasdfn939hc3, the server would send the following policy:

```
Content-Security-Policy: default-src 'self';
                         script-src 'self' https://example.com 'nonce-Nc3n83cnSAd3wc3Sasdfn939hc3'
```

Script elements can then execute either because their src URLs are whitelisted or because they have a valid nonce:

```
<script>
alert("Blocked because the policy doesn't have 'unsafe-inline'.")
</script>

<script nonce="EDNnf03nceIOfn39fn3e9h3sdfa">
alert("Still blocked because nonce is wrong.")
</script>
```

Source: http://www.w3.org/TR/CSP/

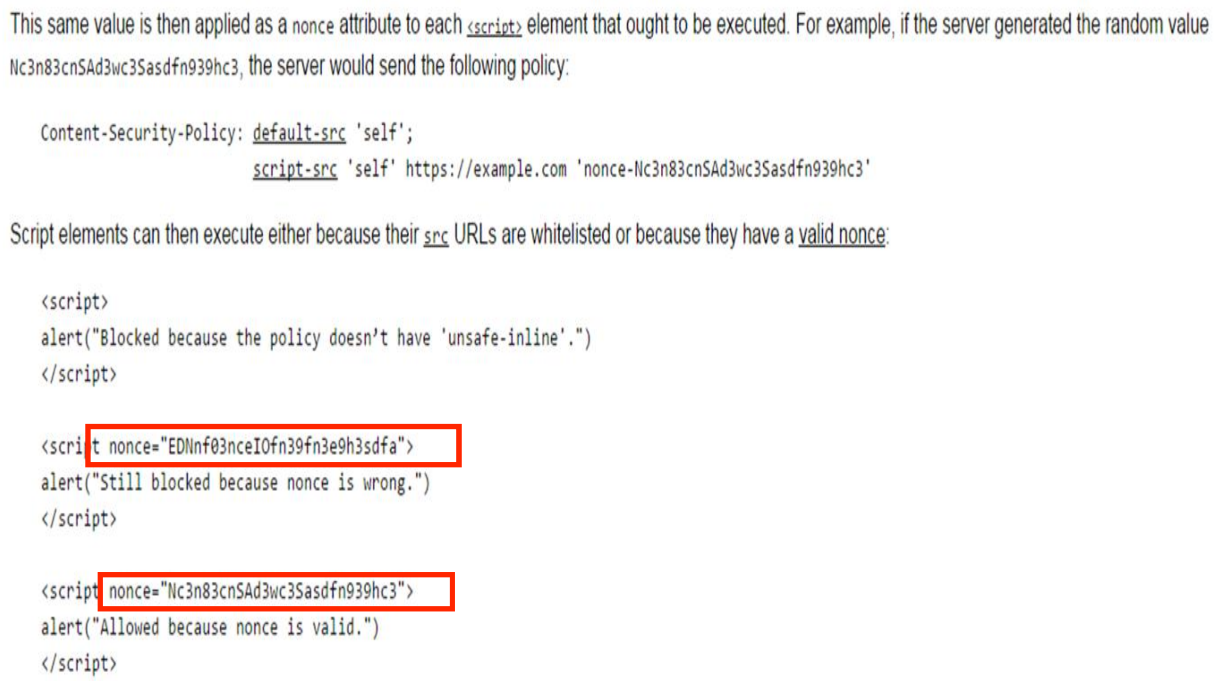
US 7712142 – Claim 25 vs. W3C standards

Claim

25. A method for document integrity, comprising: detecting an initiator insertion point in an electronic document; generating a key; associating the key with an initiator at the initiator insertion point; **associating the key with a terminator at a termination insertion point,** wherein the terminator corresponds to the initiator; and electronically transmitting the document.

W3C

As shown in the snapshot below, there are two termination insertion points. First one is when the script is invalid and here the terminator (nonce) is linked with key which does not match with the key value at the insertion point. The second one is when the script is valid and here the terminator is associated with key which matches the key value of the initiator.



This same value is then applied as a `nonce` attribute to each `<script>` element that ought to be executed. For example, if the server generated the random value `Nc3n83cnSAd3wc3Sasdfn939hc3`, the server would send the following policy:

```
Content-Security-Policy: default-src 'self';
                         script-src 'self' https://example.com 'nonce-Nc3n83cnSAd3wc3Sasdfn939hc3'
```

Script elements can then execute either because their `src` URLs are whitelisted or because they have a valid nonce:

```
<script>
alert("Blocked because the policy doesn't have 'unsafe-inline'.")
</script>

<script nonce="EDNnf03nceIOfn39fn3e9h3sdfa">
alert("Still blocked because nonce is wrong.")
</script>

<script nonce="Nc3n83cnSAd3wc3Sasdfn939hc3">
alert("Allowed because nonce is valid.")
</script>
```

Source: http://www.w3.org/TR/CSP/

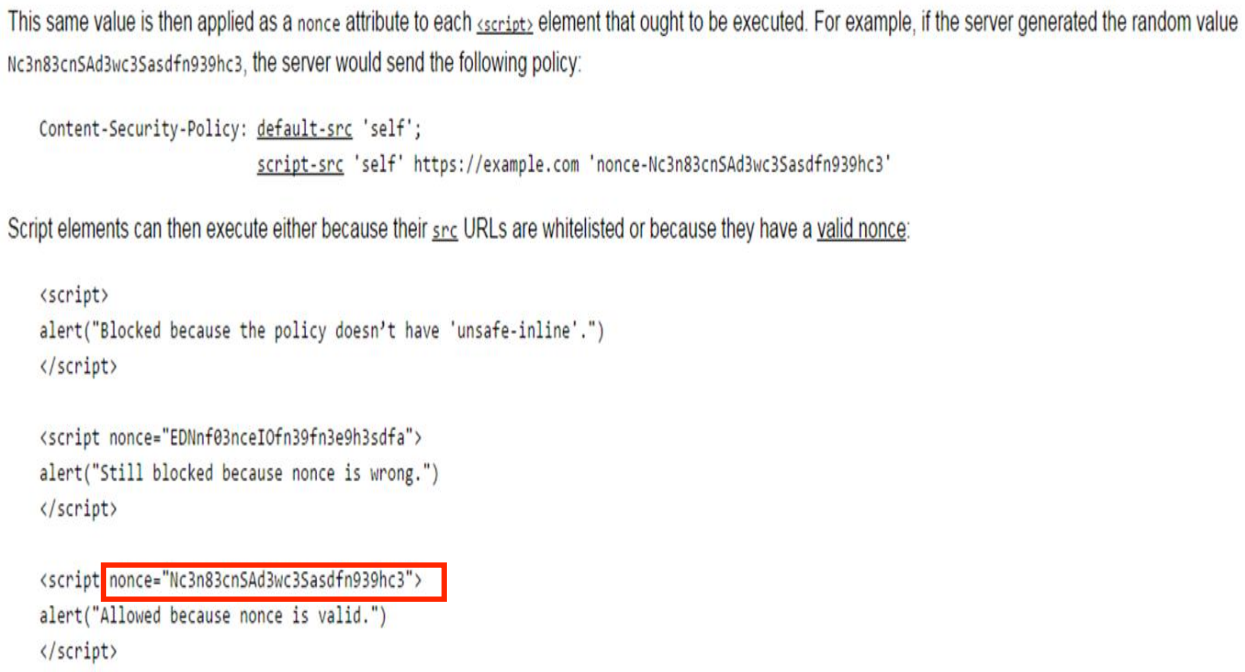
US 7712142 – Claim 25 vs. W3C standards

Claim

25. A method for document integrity, comprising: detecting an initiator insertion point in an electronic document; generating a key; associating the key with an initiator at the initiator insertion point; **associating the key with a terminator at a termination insertion point,** wherein the terminator corresponds to the initiator; and electronically transmitting the document.

W3C

As shown in the snapshot below, script elements are sent by the server with a nonce value attached to it. This nonce value can be construed to be the terminator. The nonce is associated with the key (Nc3n83cnSAd3wc3Sasdfn939hc3) when the document is valid. This terminator key value matches the key value of the initiator.



This same value is then applied as a `nonce` attribute to each `<script>` element that ought to be executed. For example, if the server generated the random value `Nc3n83cnSAd3wc3Sasdfn939hc3`, the server would send the following policy:

```
Content-Security-Policy: default-src 'self';
                         script-src 'self' https://example.com 'nonce-Nc3n83cnSAd3wc3Sasdfn939hc3'
```

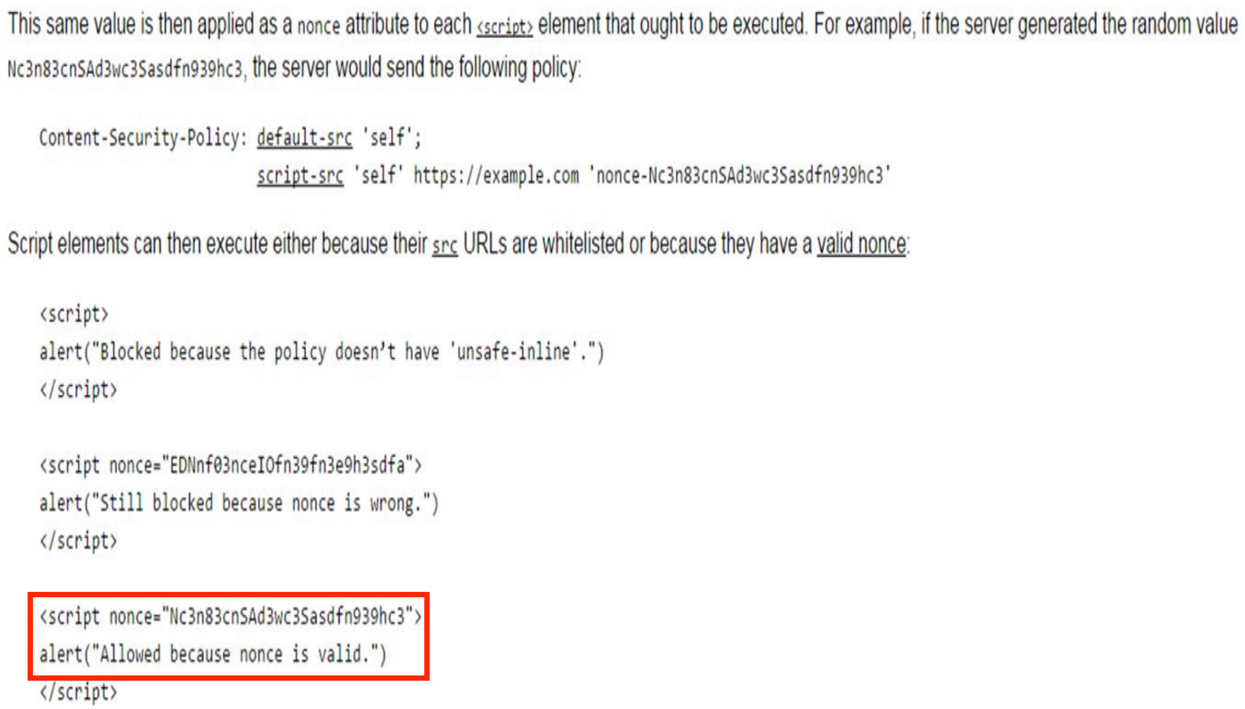
Script elements can then execute either because their `src` URLs are whitelisted or because they have a valid nonce:

```
<script>
alert("Blocked because the policy doesn't have 'unsafe-inline'.")
</script>

<script nonce="EDNnf03nceIOfn39fn3e9h3sdfa">
alert("Still blocked because nonce is wrong.")
</script>

<script nonce="Nc3n83cnSAd3wc3Sasdfn939hc3">
alert("Allowed because nonce is valid.")
</script>
```

Source: http://www.w3.org/TR/CSP/

US 7712142 – Claim 25 vs. W3C standards

Claim	W3C

25. A method for document integrity, comprising: detecting an initiator insertion point in an electronic document; generating a key; associating the key with an initiator at the initiator insertion point; associating the key with a terminator at a termination insertion point, **wherein the terminator corresponds to the initiator;** and electronically transmitting the document.	As shown in the snapshot below, the value of nonce (terminator) in the termination section is checked against the value of nonce (initiator) in the Content Security Policy header. Thus, the terminator corresponds with the initiator.
	
	This same value is then applied as a `nonce` attribute to each `<script>` element that ought to be executed. For example, if the server generated the random value `Nc3n83cnSAd3wc3Sasdfn939hc3`, the server would send the following policy:
	`Content-Security-Policy: default-src 'self';`
	`script-src 'self' https://example.com 'nonce-Nc3n83cnSAd3wc3Sasdfn939hc3'`
	Script elements can then execute either because their `src` URLs are whitelisted or because they have a valid nonce:
	`<script>`
	`alert("Blocked because the policy doesn't have 'unsafe-inline'.")`
	`</script>`
	`<script nonce="EDNnf03nceIOfn39fn3e9h3sdfa">`
	`alert("Still blocked because nonce is wrong.")`
	`</script>`
	`<script nonce="Nc3n83cnSAd3wc3Sasdfn939hc3">`
	`alert("Allowed because nonce is valid.")`
	`</script>`
	Source: http://www.w3.org/TR/CSP/

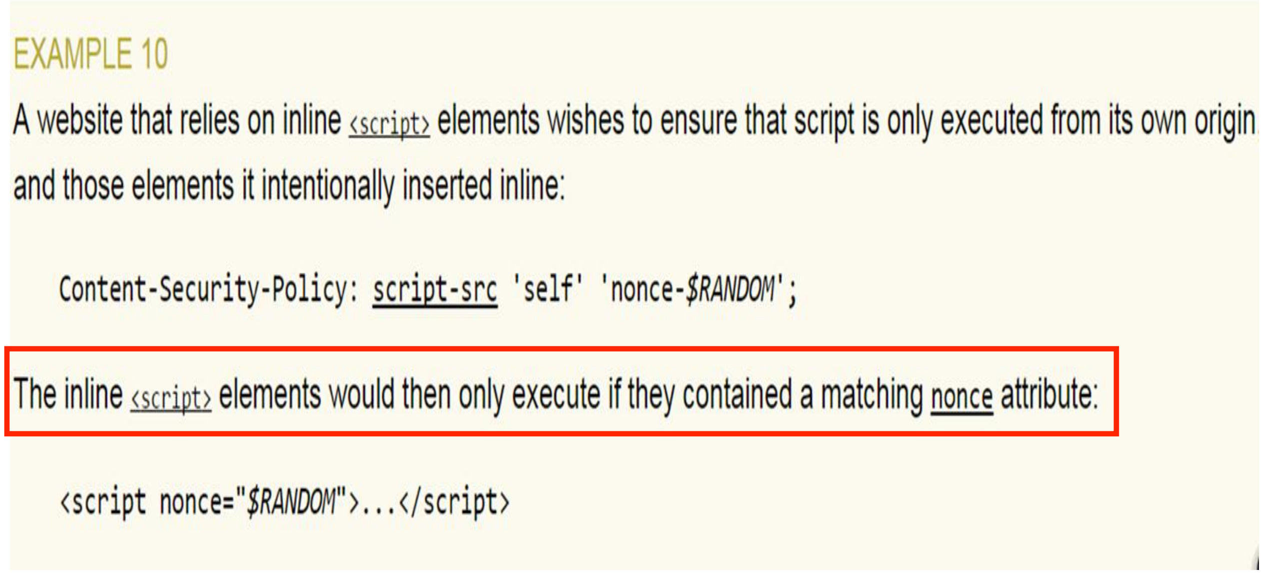
US 7712142 – Claim 25 vs. W3C standards

Claim

25. A method for document integrity, comprising: detecting an initiator insertion point in an electronic document; generating a key; associating the key with an initiator at the initiator insertion point; associating the key with a terminator at a termination insertion point, wherein the terminator corresponds to the initiator; and **electronically transmitting the document.**

W3C

As shown in the snapshot below, if the key value of the initiator and terminator matches, the script (document) is executed (transmitted).



EXAMPLE 10

A website that relies on inline <script> elements wishes to ensure that script is only executed from its own origin and those elements it intentionally inserted inline:

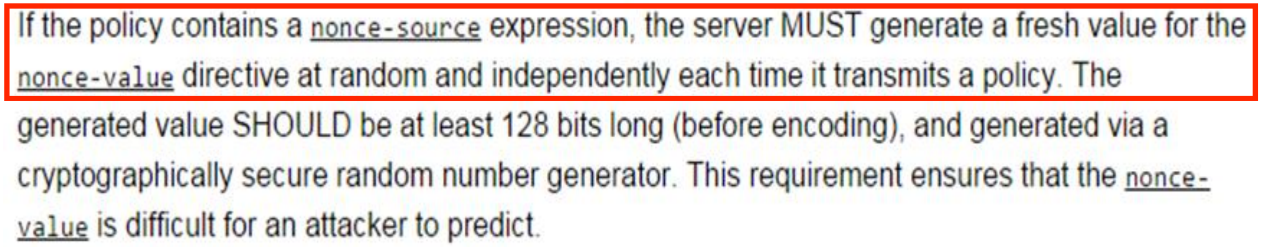
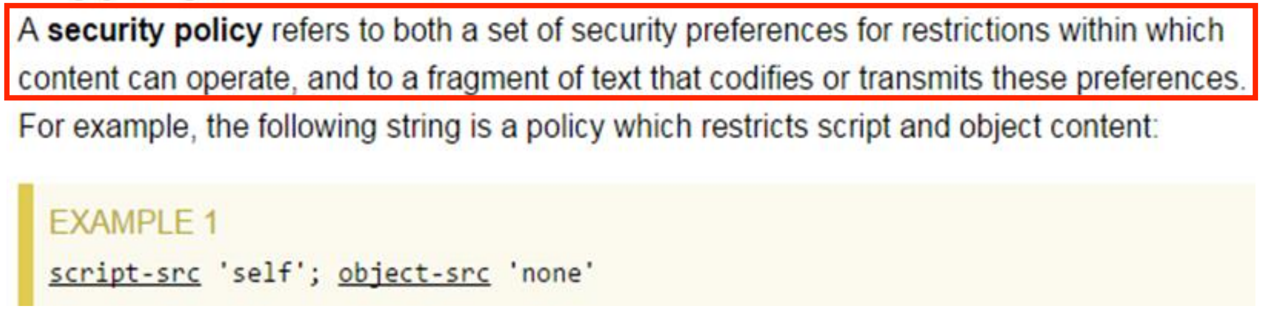
```
Content-Security-Policy: script-src 'self' 'nonce-$RANDOM';
```

The inline <script> elements would then only execute if they contained a matching nonce attribute:

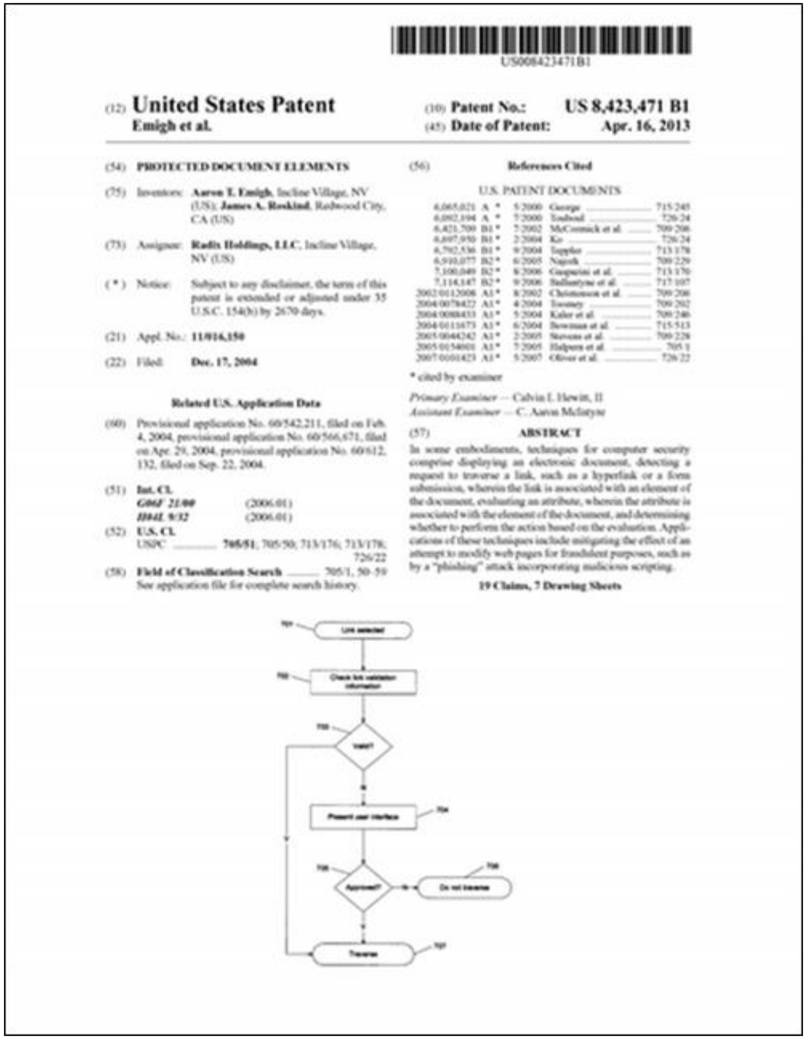
```
<script nonce="$RANDOM">...</script>
```

Source: http://www.w3.org/TR/CSP/

US 7712142 – Claim 25 vs. W3C standards

Claim	W3C
25. A method for document integrity, comprising: detecting an initiator insertion point in an electronic document; generating a key; associating the key with an initiator at the initiator insertion point; associating the key with a terminator at a termination insertion point, wherein the terminator corresponds to the initiator; and **electronically transmitting the document.**	As shown in the snapshot below, the HTML elements (document) are transmitted from the server to the client along with the policy preferences.  If the policy contains a nonce-source expression, the server MUST generate a fresh value for the nonce-value directive at random and independently each time it transmits a policy. The generated value SHOULD be at least 128 bits long (before encoding), and generated via a cryptographically secure random number generator. This requirement ensures that the nonce-value is difficult for an attacker to predict.  A **security policy** refers to both a set of security preferences for restrictions within which content can operate, and to a fragment of text that codifies or transmits these preferences. For example, the following string is a policy which restricts script and object content: EXAMPLE 1 `script-src 'self'; object-src 'none'` Source: http://www.w3.org/TR/CSP/

Family 3: US 8423471 – Bibliographic information



US008423471B1

(12) **United States Patent**
Emigh et al.

(10) **Patent No.:** **US 8,423,471 B1**
(45) **Date of Patent:** **Apr. 16, 2013**

(54) **PROTECTED DOCUMENT ELEMENTS**

(75) Inventors: **Aaron T. Emigh**, Incline Village, NV (US); **James A. Roskind**, Redwood City, CA (US)

(73) Assignee: **Radix Holdings, LLC**, Incline Village, NV (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 2670 days.

(21) Appl. No.: **11/016,150**

(22) Filed: **Dec. 17, 2004**

Related U.S. Application Data

(60) Provisional application No. 60/542,211, filed on Feb. 4, 2004, provisional application No. 60/566,671, filed on Apr. 29, 2004, provisional application No. 60/612,132, filed on Sep. 22, 2004.

(51) **Int. Cl.**
G06F 21/00 (2006.01)
H04L 9/32 (2006.01)

(52) **U.S. Cl.**
USPC **705/51**; 705/50; 713/176; 713/178; 726/22

(58) **Field of Classification Search** 705/1, 50-59
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,065,021	A	*	5/2000	George	715/245
6,092,194	A	*	7/2000	Touboul	726/24
6,421,709	B1	*	7/2002	McCormick et al.	709/206
6,697,950	B1	*	2/2004	Ko	726/24
6,792,536	B1	*	9/2004	Teppler	713/178
6,910,077	B2	*	6/2005	Najork	709/229
7,100,049	B2	*	8/2006	Gasparini et al.	713/170
7,114,147	B2	*	9/2006	Ballantyne et al.	717/107
2002/0112008	A1	*	8/2002	Christenson et al.	709/206
2004/0078422	A1	*	4/2004	Toomey	709/202
2004/0088433	A1	*	5/2004	Kaler et al.	709/246
2004/0111673	A1	*	6/2004	Bowman et al.	715/513
2005/0044242	A1	*	2/2005	Stevens et al.	709/228
2005/0154601	A1	*	7/2005	Halpern et al.	705/1
2007/0101423	A1	*	5/2007	Oliver et al.	726/22

* cited by examiner

Primary Examiner — Calvin L Hewitt, II
Assistant Examiner — C. Aaron McIntyre

(57) **ABSTRACT**

In some embodiments, techniques for computer security comprise displaying an electronic document, detecting a request to traverse a link, such as a hyperlink or a form submission, wherein the link is associated with an element of the document, evaluating an attribute, wherein the attribute is associated with the element of the document, and determining whether to perform the action based on the evaluation. Applications of these techniques include mitigating the effect of an attempt to modify web pages for fraudulent purposes, such as by a "phishing" attack incorporating malicious scripting.

19 Claims, 7 Drawing Sheets

Patent of Interest:

US8423471
(Priority date: Feb 04, 2004)

Protected document elements

Exemplary Market Applications:

The patented technology finds applications in web security.

US 8423471 – Claim 8

8. A non-transitory computer readable medium storing computer instructions which when executed by a processor cause the processor to perform the steps of:

   displaying an electronic document;

   detecting a request to traverse a link, wherein the link is associated with an element of the document;

   evaluating an attribute, wherein the attribute is associated with the element of the document and wherein evaluating the attribute includes determining whether a destination associated with the link was enumerated; and

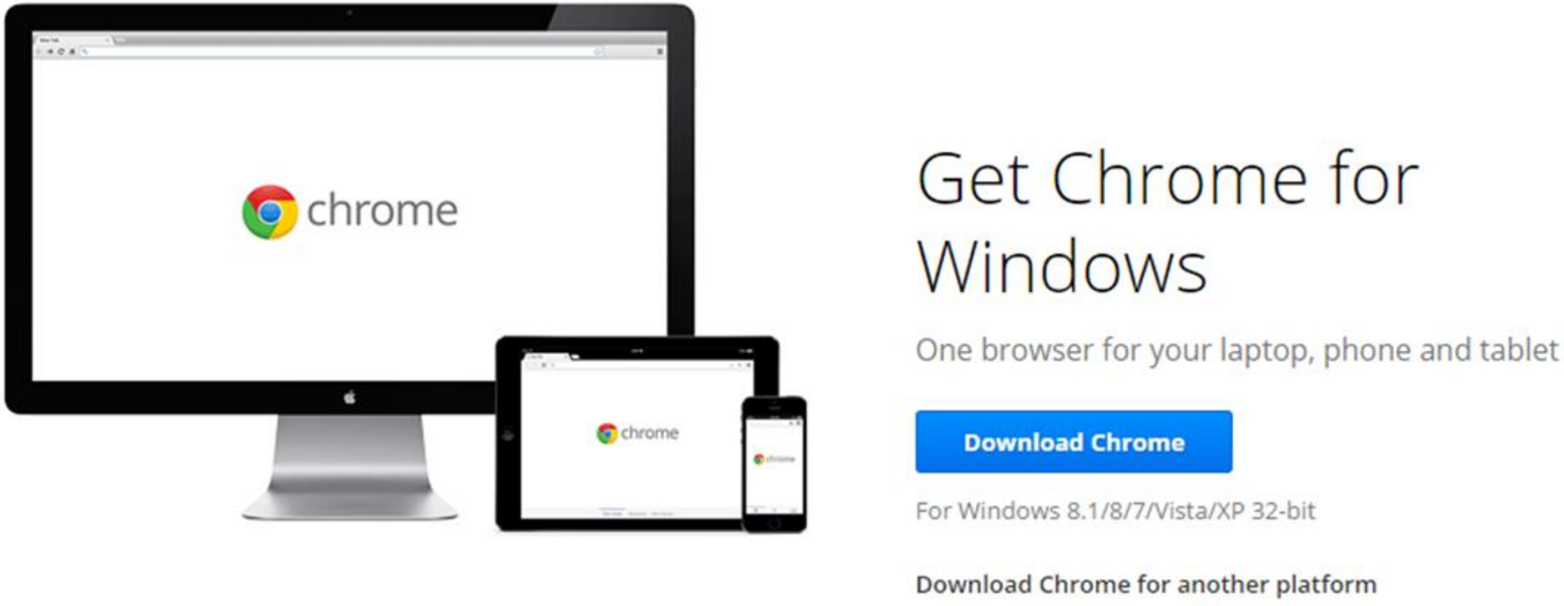
   determining whether to traverse the link based on the evaluation.

US 8423471 – EoU Summary

Google	
Key claim(s)	1, 7, and 8(independent claims)
Mapped product	Chart has been made with respect to Google Chrome Browser. http://www.google.com/chrome/
Source	Information related to product is available at: https://support.google.com/chrome/answer/99020 https://www.google.com/intl/en/chrome/browser/features.html#security https://developers.google.com/safe-browsing/ https://developer.chrome.com/devtools/docs/dom-and-styles
Product launch date	September 2, 2008
Details of product	Google Chrome is a web browser by Google Inc. The browser includes safe browsing option which prevents the computer from accessing the websites that contain malwares. Also, it protects users from phishing.

US 8423471 – Google Chrome: Overview

Google Chrome is a web browser by Google Inc. The browser includes safe browsing option which prevents the computer from accessing the websites that contain malwares. Also, it protects users from phishing.



Get Chrome for Windows

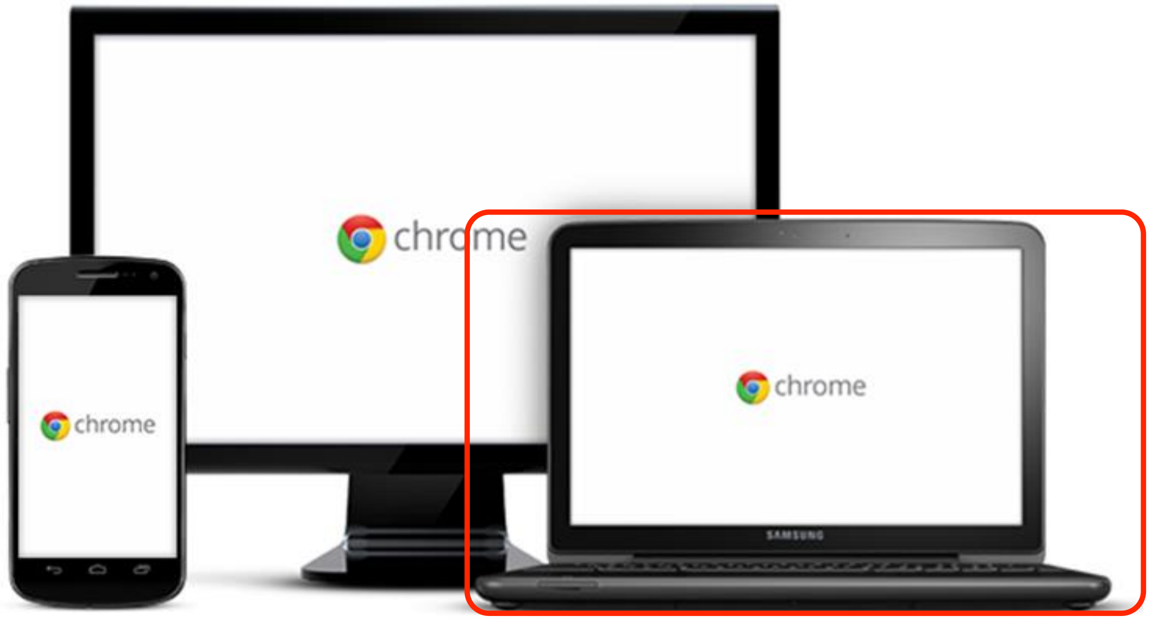
One browser for your laptop, phone and tablet

For Windows 8.1/8/7/Vista/XP 32-bit

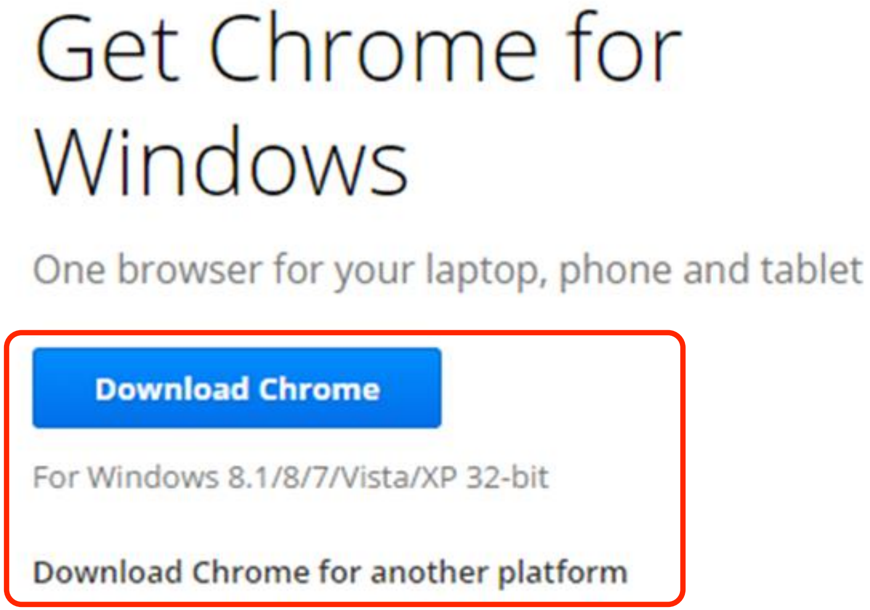
Download Chrome for another platform

Source: https://www.google.com/chrome/browser/desktop/index.html

US 8423471 – Google Chrome: Overview

Claim	Google
8. **A non-transitory computer readable medium storing computer instructions which when executed by a processor cause the processor to perform the steps of:** displaying an electronic document; detecting a request to traverse a link, wherein the link is associated with an element of the document; evaluating an attribute, wherein the attribute is associated with the element of the document and wherein evaluating the attribute includes determining whether a destination associated with the link was enumerated; and determining whether to traverse the link based on the evaluation.	As shown in the snapshot below, Google Chrome is installed in the personal computers, laptops, and mobile devices. The electronic devices on which the Google Chrome is installed includes a non-transitory computer readable medium (RAM).  Source: https://www.google.com/chrome/browser/signin.html

US 8423471 – Claim 8 vs. Google Chrome

Claim	Google
8. **A non-transitory computer readable medium storing computer instructions which when executed by a processor cause the processor to perform the steps of:** displaying an electronic document; detecting a request to traverse a link, wherein the link is associated with an element of the document; evaluating an attribute, wherein the attribute is associated with the element of the document and wherein evaluating the attribute includes determining whether a destination associated with the link was enumerated; and determining whether to traverse the link based on the evaluation.	As shown in the snapshot below, Google Chrome software includes the computer instructions. Further, the computer instructions are executed by the processor. The processor is included in the electronic device in which Google Chrome is installed.  Get Chrome for Windows One browser for your laptop, phone and tablet Download Chrome For Windows 8.1/8/7/Vista/XP 32-bit Download Chrome for another platform Source: https://www.google.com/chrome/browser/desktop/index.html

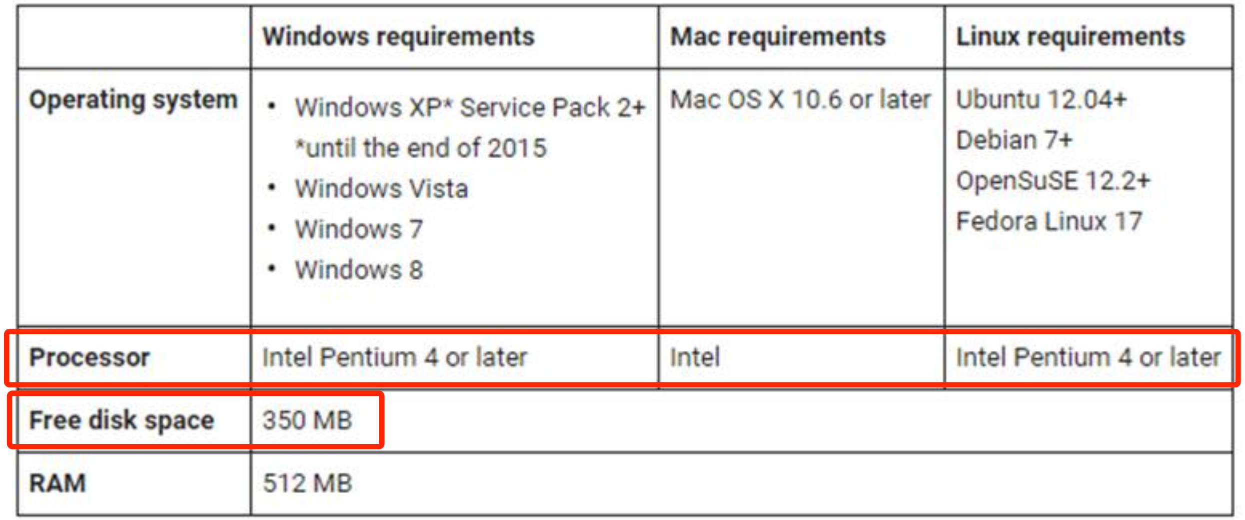
US 8423471 – Claim 8 vs. Google Chrome

Claim

8. **A non-transitory computer readable medium storing computer instructions which when executed by a processor cause the processor to perform the steps of:** displaying an electronic document; detecting a request to traverse a link, wherein the link is associated with an element of the document; evaluating an attribute, wherein the attribute is associated with the element of the document and wherein evaluating the attribute includes determining whether a destination associated with the link was enumerated; and determining whether to traverse the link based on the evaluation.

Google

As shown in the snapshot below, Google Chrome has specific system requirements of free disk space and processor. Further, Google Chrome includes computer instructions which are executed by a processor of the electronic device (personal computer, laptop, and mobile device)



	Windows requirements	Mac requirements	Linux requirements
Operating system	• Windows XP* Service Pack 2+ *until the end of 2015 • Windows Vista • Windows 7 • Windows 8	Mac OS X 10.6 or later	Ubuntu 12.04+ Debian 7+ OpenSuSE 12.2+ Fedora Linux 17
Processor	Intel Pentium 4 or later	Intel	Intel Pentium 4 or later
Free disk space	350 MB		
RAM	512 MB		

Source: https://support.google.com/chrome/answer/95346?hl=en

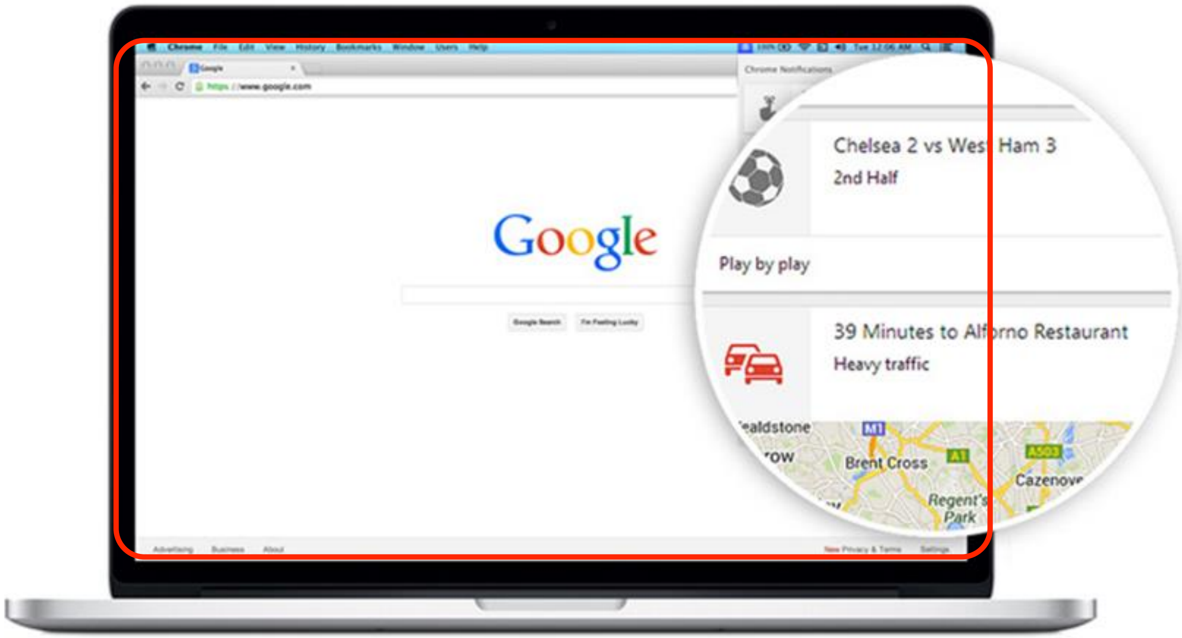
US 8423471 – Claim 8 vs. Google Chrome

Claim

8. A non-transitory computer readable medium storing computer instructions which when executed by a processor cause the processor to perform the steps of: **displaying an electronic document;** detecting a request to traverse a link, wherein the link is associated with an element of the document; evaluating an attribute, wherein the attribute is associated with the element of the document and wherein evaluating the attribute includes determining whether a destination associated with the link was enumerated; and determining whether to traverse the link based on the evaluation.

Google

As shown in the snapshot below, Google Chrome displays an electronic document. The electronic document is the HTML/web page.



Source: https://www.google.com/chrome/browser/desktop/index.html

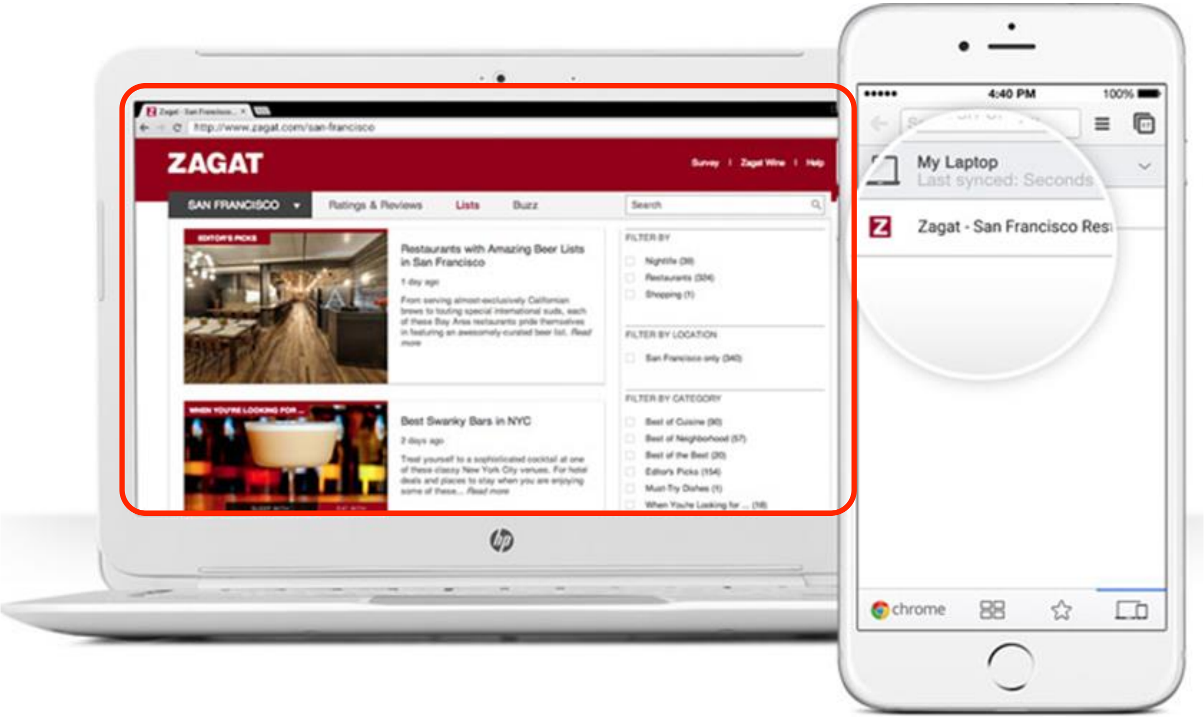
US 8423471 – Claim 8 vs. Google Chrome

Claim

8. A non-transitory computer readable medium storing computer instructions which when executed by a processor cause the processor to perform the steps of: **displaying an electronic document;** detecting a request to traverse a link, wherein the link is associated with an element of the document; evaluating an attribute, wherein the attribute is associated with the element of the document and wherein evaluating the attribute includes determining whether a destination associated with the link was enumerated; and determining whether to traverse the link based on the evaluation.

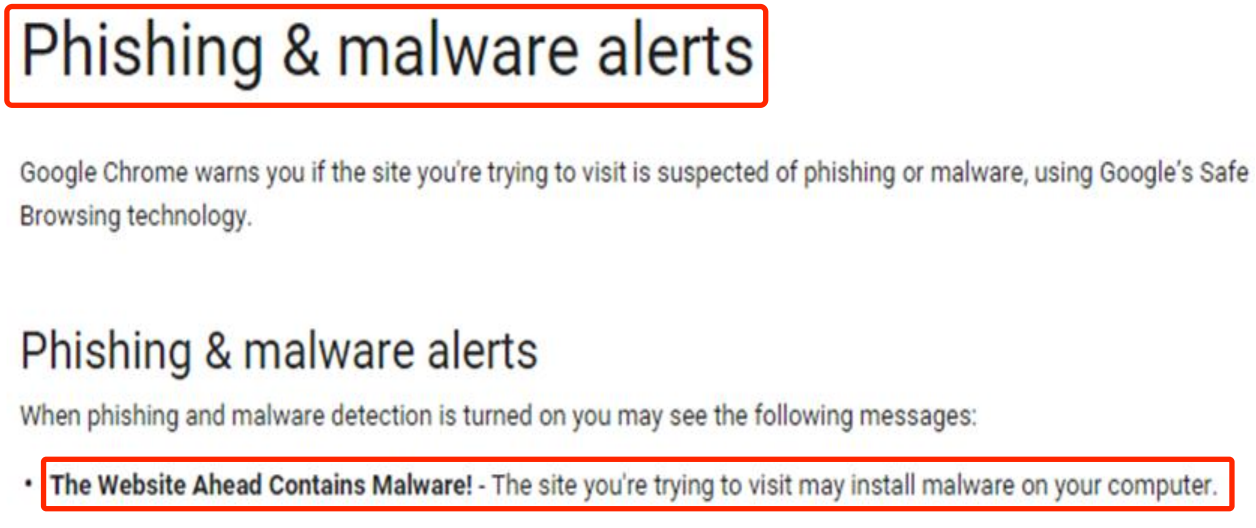
Google

As shown in the snapshot below, Google Chrome displays an electronic document. The electronic document is the HTML/web page.

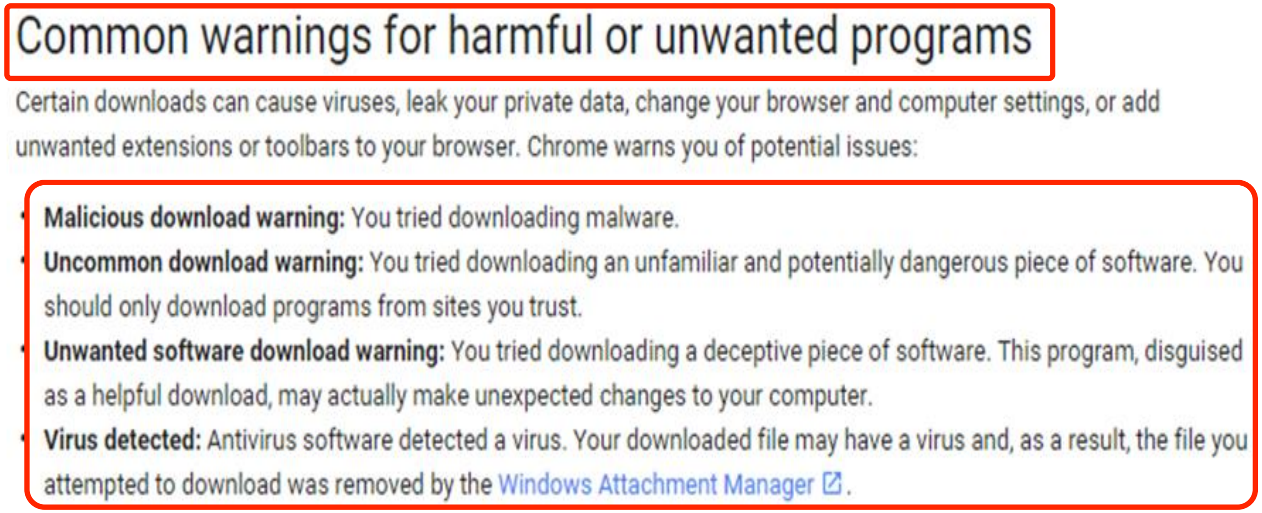


Source: https://www.google.com/chrome/browser/desktop/index.html

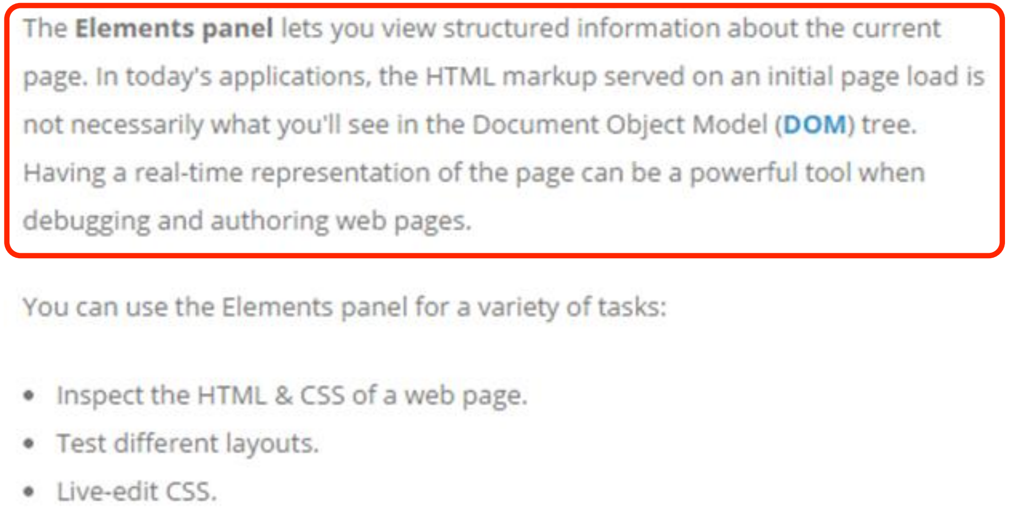
US 8423471 – Claim 8 vs. Google Chrome

Claim	Google
8. A non-transitory computer readable medium storing computer instructions which when executed by a processor cause the processor to perform the steps of: displaying an electronic document; **detecting a request to traverse a link, wherein the link is associated with an element of the document;** evaluating an attribute, wherein the attribute is associated with the element of the document and wherein evaluating the attribute includes determining whether a destination associated with the link was enumerated; and determining whether to traverse the link based on the evaluation.	As shown in the snapshot below, Google Chrome detects a request to traverse a hyperlink included in the web page. The request is made by the user when the user clicks the hyperlink. As the user clicks the hyperlink, phishing and malware alerts are generated if the website is suspicious. Further, the hyperlink included in the web page is a part of the electronic document (web page). Thus, Google Chrome detects a request to traverse a link, wherein the link is associated with the element of the document.  Phishing & malware alerts Google Chrome warns you if the site you're trying to visit is suspected of phishing or malware, using Google's Safe Browsing technology. Phishing & malware alerts When phishing and malware detection is turned on you may see the following messages: • **The Website Ahead Contains Malware!** - The site you're trying to visit may install malware on your computer. Source: https://support.google.com/chrome/answer/99020?hl=en

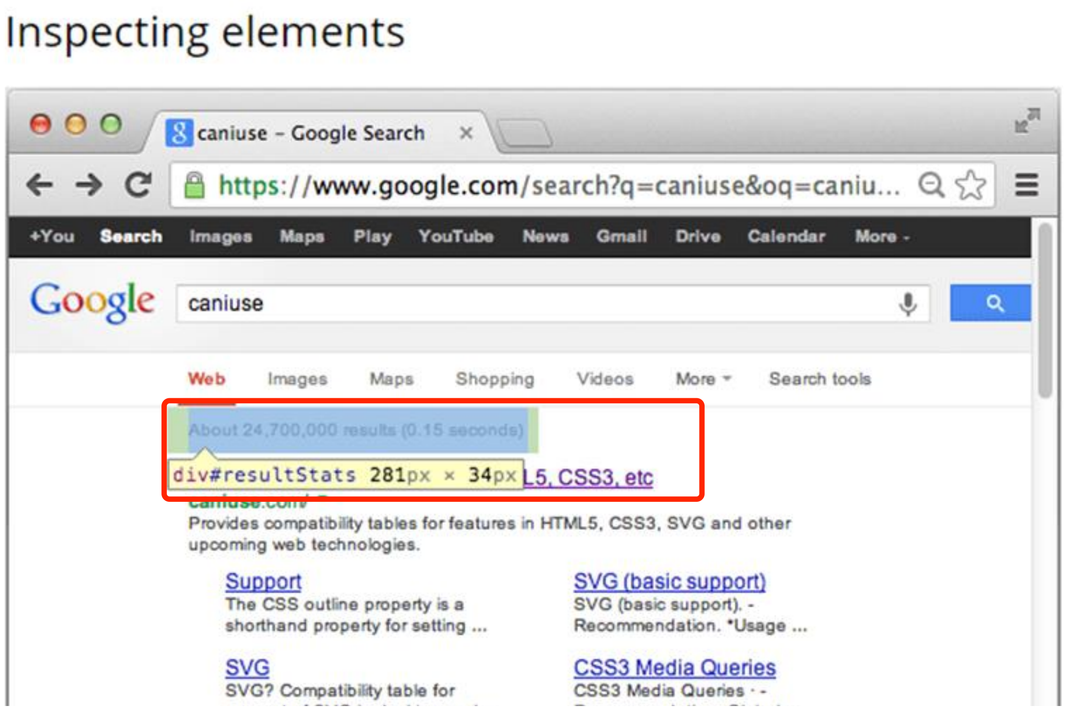
US 8423471 – Claim 8 vs. Google Chrome

Claim	Google
8. A non-transitory computer readable medium storing computer instructions which when executed by a processor cause the processor to perform the steps of: displaying an electronic document; **detecting a request to traverse a link, wherein the link is associated with an element of the document;** evaluating an attribute, wherein the attribute is associated with the element of the document and wherein evaluating the attribute includes determining whether a destination associated with the link was enumerated; and determining whether to traverse the link based on the evaluation.	As shown in the snapshot below, Google Chrome provides warning when the browser attempts to download from the suspicious web sites. This request is detected by the Google Chrome browser. The request includes a link from which the download is carried out.  Common warnings for harmful or unwanted programs Certain downloads can cause viruses, leak your private data, change your browser and computer settings, or add unwanted extensions or toolbars to your browser. Chrome warns you of potential issues: - **Malicious download warning:** You tried downloading malware. - **Uncommon download warning:** You tried downloading an unfamiliar and potentially dangerous piece of software. You should only download programs from sites you trust. - **Unwanted software download warning:** You tried downloading a deceptive piece of software. This program, disguised as a helpful download, may actually make unexpected changes to your computer. - **Virus detected:** Antivirus software detected a virus. Your downloaded file may have a virus and, as a result, the file you attempted to download was removed by the Windows Attachment Manager. Source: https://support.google.com/chrome/answer/2898334?hl=en

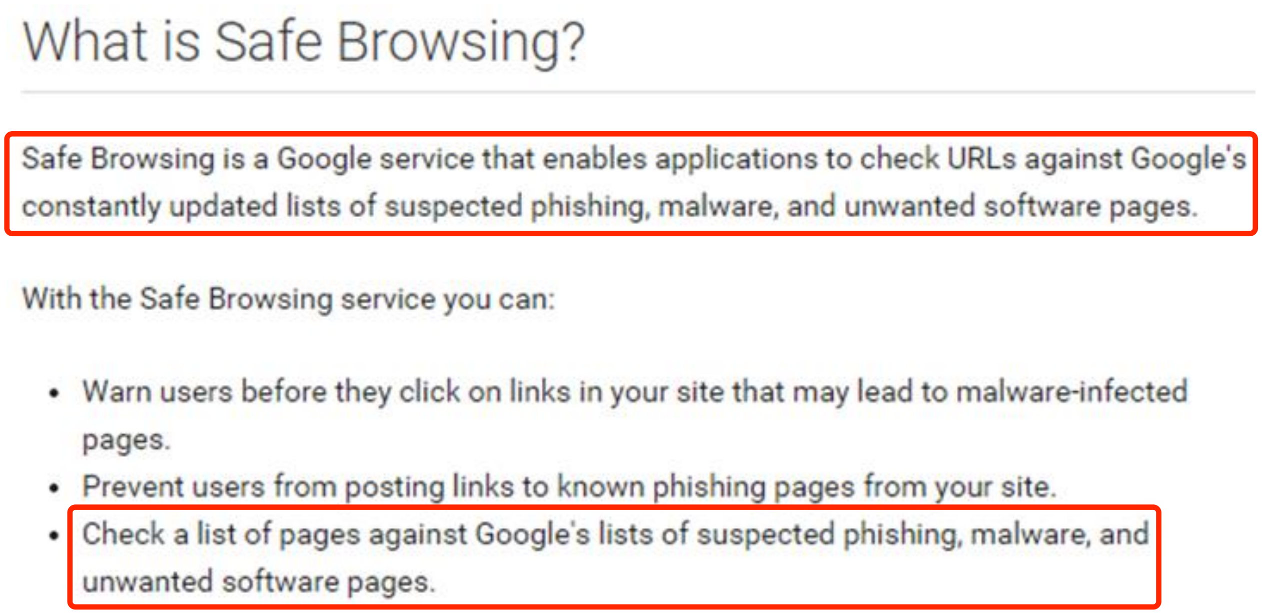
US 8423471 – Claim 8 vs. Google Chrome

Claim	Google
8. A non-transitory computer readable medium storing computer instructions which when executed by a processor cause the processor to perform the steps of: displaying an electronic document; **detecting a request to traverse a link, wherein the link is associated with an element of the document;** evaluating an attribute, wherein the attribute is associated with the element of the document and wherein evaluating the attribute includes determining whether a destination associated with the link was enumerated; and determining whether to traverse the link based on the evaluation.	As shown in the snapshot below, web pages include elements. The hyperlink included in the webpage (electronic document) is associated with the element of the document object model.  The **Elements panel** lets you view structured information about the current page. In today's applications, the HTML markup served on an initial page load is not necessarily what you'll see in the Document Object Model (**DOM**) tree. Having a real-time representation of the page can be a powerful tool when debugging and authoring web pages. You can use the Elements panel for a variety of tasks: • Inspect the HTML & CSS of a web page. • Test different layouts. • Live-edit CSS. Source: https://developer.chrome.com/devtools/docs/dom-and-styles

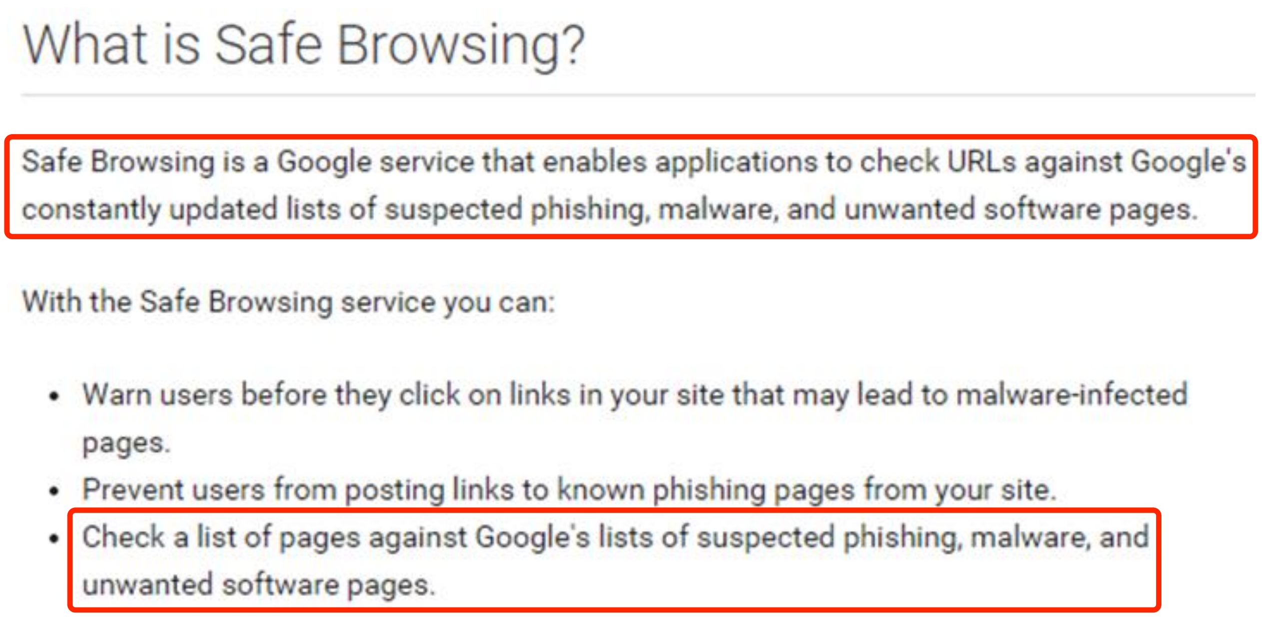
US 8423471 – Claim 8 vs. Google Chrome

Claim	Google
8. A non-transitory computer readable medium storing computer instructions which when executed by a processor cause the processor to perform the steps of: displaying an electronic document; **detecting a request to traverse a link, wherein the link is associated with an element of the document;** evaluating an attribute, wherein the attribute is associated with the element of the document and wherein evaluating the attribute includes determining whether a destination associated with the link was enumerated; and determining whether to traverse the link based on the evaluation.	As shown in the snapshot below, web pages include elements. The hyperlink included in the webpage (electronic document) is associated with the element of the document object model.  Inspecting elements caniuse - Google Search https://www.google.com/search?q=caniuse&oq=caniu... caniuse About 24,700,000 results (0.15 seconds) div#resultStats 281px × 34px L5, CSS3, etc caniuse.com/ Provides compatibility tables for features in HTML5, CSS3, SVG and other upcoming web technologies. Support The CSS outline property is a shorthand property for setting ... SVG (basic support) SVG (basic support). - Recommendation. *Usage ... SVG SVG? Compatibility table for CSS3 Media Queries CSS3 Media Queries · - +You Search Images Maps Play YouTube News Gmail Drive Calendar More - Google Web Images Maps Shopping Videos More Search tools Source: https://developer.chrome.com/devtools/docs/dom-and-styles

US 8423471 – Claim 8 vs. Google Chrome

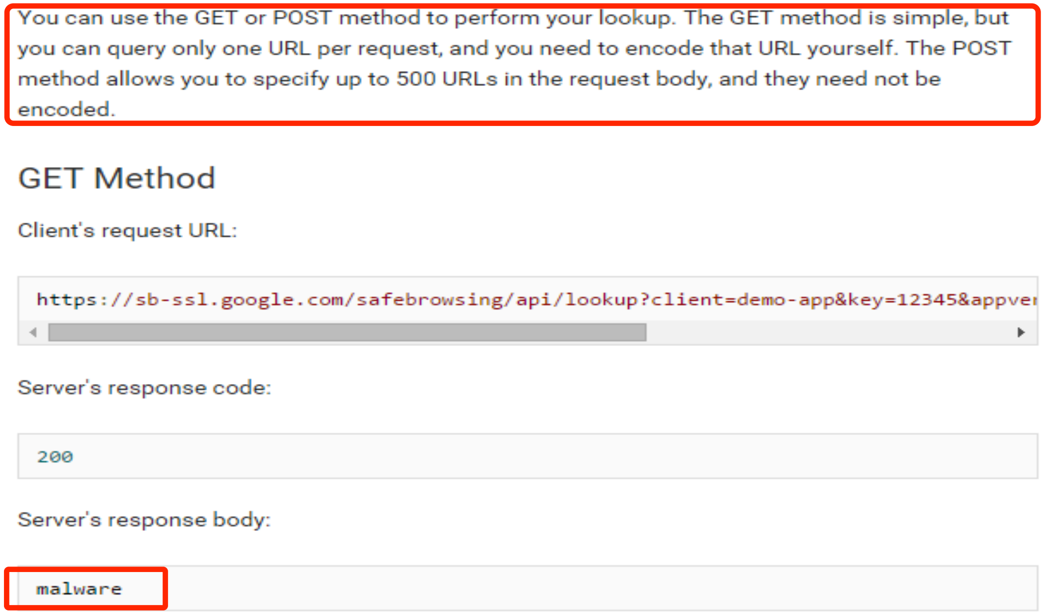
Claim	Google
8. A non-transitory computer readable medium storing computer instructions which when executed by a processor cause the processor to perform the steps of: displaying an electronic document; detecting a request to traverse a link, wherein the link is associated with an element of the document; **evaluating an attribute, wherein the attribute is associated with the element of the document** and wherein evaluating the attribute includes determining whether a destination associated with the link was enumerated; and determining whether to traverse the link based on the evaluation.	As shown in the snapshot below, Google Chrome evaluates the safety attribute (malware or phishing websites) related to requested link. The link is associated with the element of the document. Thus, Google Chrome evaluates safety attributes associated with the element of the document.  What is Safe Browsing? Safe Browsing is a Google service that enables applications to check URLs against Google's constantly updated lists of suspected phishing, malware, and unwanted software pages. With the Safe Browsing service you can: • Warn users before they click on links in your site that may lead to malware-infected pages. • Prevent users from posting links to known phishing pages from your site. • Check a list of pages against Google's lists of suspected phishing, malware, and unwanted software pages. Source: https://developers.google.com/safe-browsing/

US 8423471 – Claim 8 vs. Google Chrome

Claim	Google
8. A non-transitory computer readable medium storing computer instructions which when executed by a processor cause the processor to perform the steps of: displaying an electronic document; detecting a request to traverse a link, wherein the link is associated with an element of the document; evaluating an attribute, wherein the attribute is associated with the element of the document **and wherein evaluating the attribute includes determining whether a destination associated with the link was enumerated;** and determining whether to traverse the link based on the evaluation.	As shown in the snapshot below, Google Chrome evaluates the safety attribute of the hyperlink included in the web page. The safe browsing option of the Google Chrome enables the checking of the URL against the Google's list of suspected websites.  What is Safe Browsing? Safe Browsing is a Google service that enables applications to check URLs against Google's constantly updated lists of suspected phishing, malware, and unwanted software pages. With the Safe Browsing service you can: • Warn users before they click on links in your site that may lead to malware-infected pages. • Prevent users from posting links to known phishing pages from your site. • Check a list of pages against Google's lists of suspected phishing, malware, and unwanted software pages. Source: https://developers.google.com/safe-browsing/

US 8423471 – Claim 8 vs. Google Chrome

Claim	Google

8. A non-transitory computer readable medium storing computer instructions which when executed by a processor cause the processor to perform the steps of: displaying an electronic document; detecting a request to traverse a link, wherein the link is associated with an element of the document; evaluating an attribute, wherein the attribute is associated with the element of the document **and wherein evaluating the attribute includes determining whether a destination associated with the link was enumerated;** and determining whether to traverse the link based on the evaluation.	As shown in the snapshot below, As shown in the snapshot below, Google Chrome API look up guide includes a GET request. The GET request checks whether the requested hyperlink is enumerated in the Google's server. The server responds as per the evaluation of the attribute. Thus, Google Chrome evaluates the safety attribute by determining whether the destination (URL/web page) associated with the hyperlink was enumerated.
	
	You can use the GET or POST method to perform your lookup. The GET method is simple, but you can query only one URL per request, and you need to encode that URL yourself. The POST method allows you to specify up to 500 URLs in the request body, and they need not be encoded.
	GET Method
	Client's request URL:
	`https://sb-ssl.google.com/safebrowsing/api/lookup?client=demo-app&key=12345&appver`
	Server's response code:
	`200`
	Server's response body:
	`malware`
	Source: https://developers.google.com/safe-browsing/lookup_guide

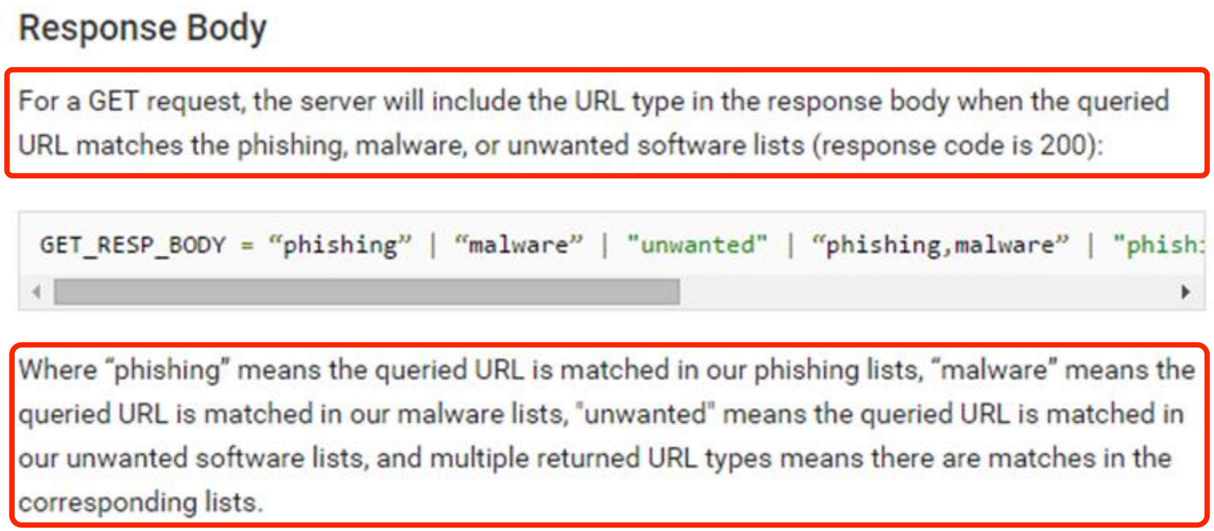
US 8423471 – Claim 8 vs. Google Chrome

Claim

8. A non-transitory computer readable medium storing computer instructions which when executed by a processor cause the processor to perform the steps of: displaying an electronic document; detecting a request to traverse a link, wherein the link is associated with an element of the document; evaluating an attribute, wherein the attribute is associated with the element of the document **and wherein evaluating the attribute includes determining whether a destination associated with the link was enumerated;** and determining whether to traverse the link based on the evaluation.

Google

As shown in the snapshot below, Google Chrome API look up guide includes a GET request. The GET request checks whether the requested hyperlink is enumerated in the Google's server (phishing, malware or unwanted software lists). Thus, Google Chrome evaluates the safety attribute by determining whether the destination (URL/web page) associated with the hyperlink was enumerated.



Response Body

For a GET request, the server will include the URL type in the response body when the queried URL matches the phishing, malware, or unwanted software lists (response code is 200):

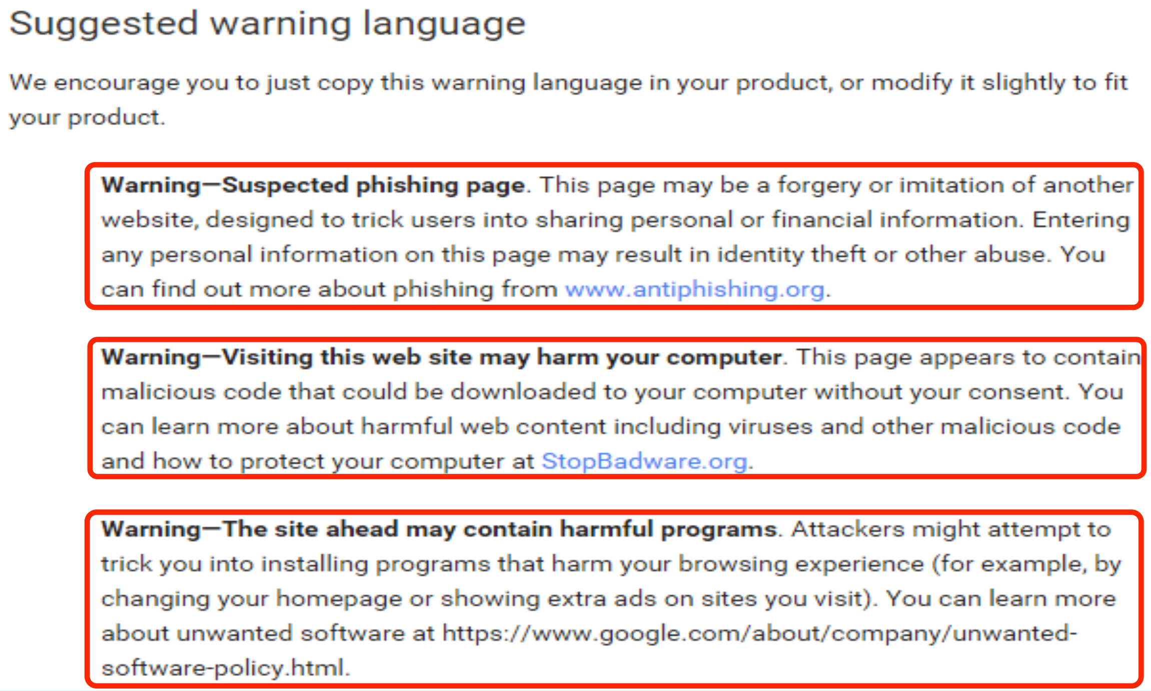
```
GET_RESP_BODY = “phishing” | “malware” | "unwanted" | “phishing,malware” | "phish:
```

Where “phishing” means the queried URL is matched in our phishing lists, “malware” means the queried URL is matched in our malware lists, "unwanted" means the queried URL is matched in our unwanted software lists, and multiple returned URL types means there are matches in the corresponding lists.

Source: https://developers.google.com/safe-browsing/lookup_guide

US 8423471 – Claim 8 vs. Google Chrome

Claim	Google
8. A non-transitory computer readable medium storing computer instructions which when executed by a processor cause the processor to perform the steps of: displaying an electronic document; detecting a request to traverse a link, wherein the link is associated with an element of the document; evaluating an attribute, wherein the attribute is associated with the element of the document and wherein evaluating the attribute includes determining whether a destination associated with the link was enumerated; **and determining whether to traverse the link based on the evaluation.**	As shown in the snapshot below, Google Chrome determines whether to traverse the link based on the evaluation. Google Chrome allows the user to traverse the link if the evaluation states that the website is safe to visit. If the link is included in the malware, phishing or unwanted software list, Google Chrome provides a warning before traversing the link. Thus, Google Chrome determines whether to traverse the link based on the evaluation.



Suggested warning language

We encourage you to just copy this warning language in your product, or modify it slightly to fit your product.

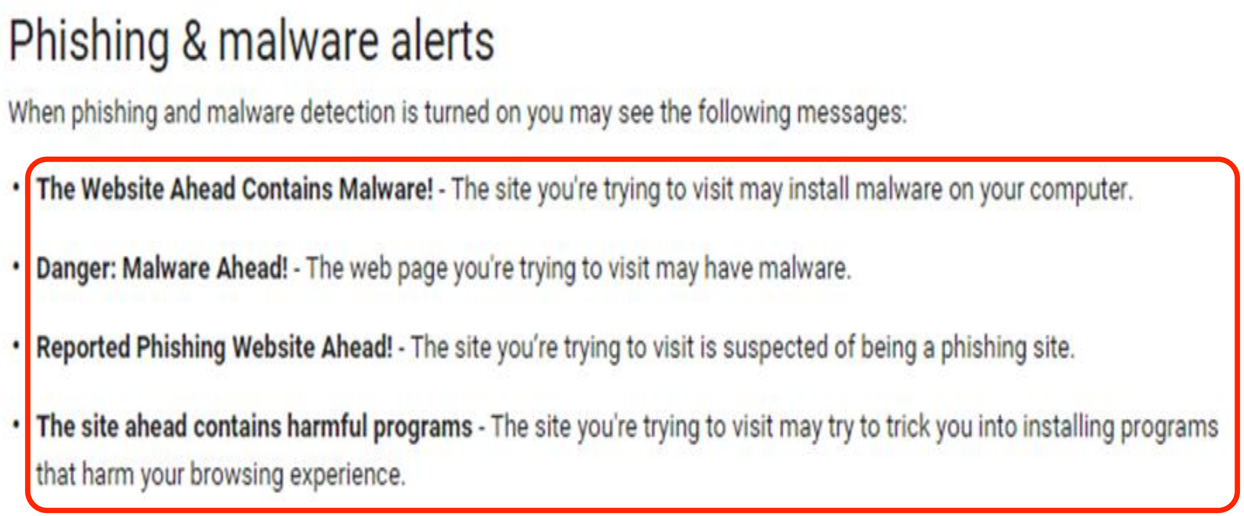
Warning—Suspected phishing page. This page may be a forgery or imitation of another website, designed to trick users into sharing personal or financial information. Entering any personal information on this page may result in identity theft or other abuse. You can find out more about phishing from www.antiphishing.org.

Warning—Visiting this web site may harm your computer. This page appears to contain malicious code that could be downloaded to your computer without your consent. You can learn more about harmful web content including viruses and other malicious code and how to protect your computer at StopBadware.org.

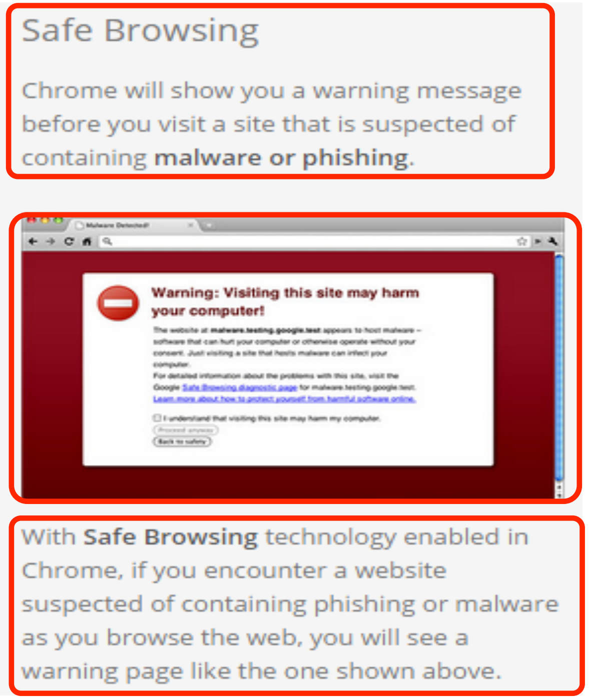
Warning—The site ahead may contain harmful programs. Attackers might attempt to trick you into installing programs that harm your browsing experience (for example, by changing your homepage or showing extra ads on sites you visit). You can learn more about unwanted software at https://www.google.com/about/company/unwanted-software-policy.html.

Source: https://developers.google.com/safe-browsing/lookup_guide#UsageRestrictions

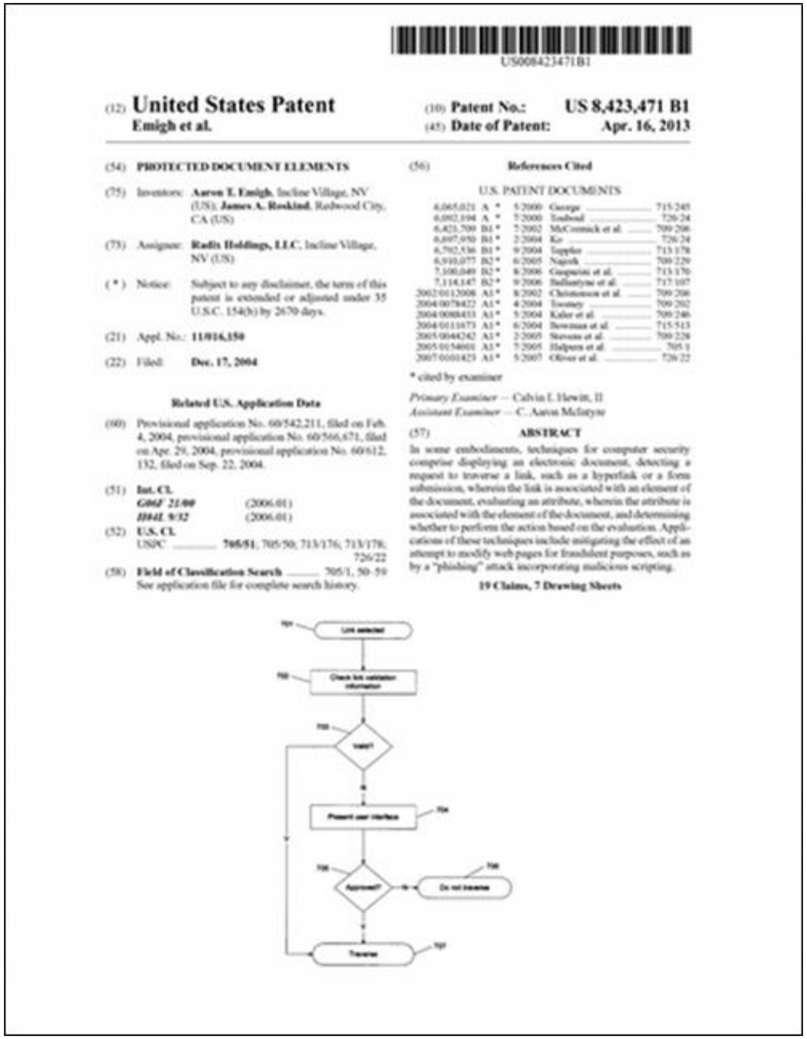
US 8423471 – Claim 8 vs. Google Chrome

Claim	Google
8. A non-transitory computer readable medium storing computer instructions which when executed by a processor cause the processor to perform the steps of: displaying an electronic document; detecting a request to traverse a link, wherein the link is associated with an element of the document; evaluating an attribute, wherein the attribute is associated with the element of the document and wherein evaluating the attribute includes determining whether a destination associated with the link was enumerated; **and determining whether to traverse the link based on the evaluation.**	As shown in the snapshot below, Google Chrome determines whether to traverse the link based on the evaluation. Google Chrome allows the user to traverse the link if the evaluation states that the website is safe to visit. If the link is included in the malware, phishing or unwanted software list, Google Chrome provides a warning before traversing the link. Thus, Google Chrome determines whether to traverse the link based on the evaluation.  Phishing & malware alerts When phishing and malware detection is turned on you may see the following messages: - **The Website Ahead Contains Malware!** - The site you're trying to visit may install malware on your computer. - **Danger: Malware Ahead!** - The web page you're trying to visit may have malware. - **Reported Phishing Website Ahead!** - The site you're trying to visit is suspected of being a phishing site. - **The site ahead contains harmful programs** - The site you're trying to visit may try to trick you into installing programs that harm your browsing experience. Source: https://support.google.com/chrome/answer/99020?hl=en

US 8423471 – Claim 8 vs. Google Chrome

Claim	Google
8. A non-transitory computer readable medium storing computer instructions which when executed by a processor cause the processor to perform the steps of: displaying an electronic document; detecting a request to traverse a link, wherein the link is associated with an element of the document; evaluating an attribute, wherein the attribute is associated with the element of the document and wherein evaluating the attribute includes determining whether a destination associated with the link was enumerated; **and determining whether to traverse the link based on the evaluation.**	As shown in the snapshot below, Google Chrome allows the user to traverse the link if the evaluation states that the website is safe to visit. If the link is included in the malware, phishing or unwanted software list, Google Chrome provides a warning before traversing the link. Thus, Google Chrome determines whether to traverse the link based on the evaluation.  Safe Browsing Chrome will show you a warning message before you visit a site that is suspected of containing **malware or phishing**. Malware Detected! **Warning: Visiting this site may harm your computer!** The website at **malware.testing.google.test** appears to host malware – software that can hurt your computer or otherwise operate without your consent. Just visiting a site that hosts malware can infect your computer. For detailed information about the problems with this site, visit the Google Safe Browsing diagnostic page for malware.testing.google.test. Learn more about how to protect yourself from harmful software online. □ I understand that visiting this site may harm my computer. Proceed anyway Back to safety With **Safe Browsing** technology enabled in Chrome, if you encounter a website suspected of containing phishing or malware as you browse the web, you will see a warning page like the one shown above. Source: https://www.google.com/intl/en/chrome/browser/features.html#security

US 8423471 – Bibliographic information



US008423471B1

(12) **United States Patent**
Emigh et al.

(10) **Patent No.:** **US 8,423,471 B1**
(45) **Date of Patent:** **Apr. 16, 2013**

(54) **PROTECTED DOCUMENT ELEMENTS**

(75) Inventors: **Aaron T. Emigh**, Incline Village, NV (US); **James A. Roskind**, Redwood City, CA (US)

(73) Assignee: **Radix Holdings, LLC**, Incline Village, NV (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 2670 days.

(21) Appl. No.: **11/016,150**

(22) Filed: **Dec. 17, 2004**

Related U.S. Application Data

(60) Provisional application No. 60/542,211, filed on Feb. 4, 2004, provisional application No. 60/566,671, filed on Apr. 29, 2004, provisional application No. 60/612,132, filed on Sep. 22, 2004.

(51) **Int. Cl.**
G06F 21/00 (2006.01)
H04L 9/32 (2006.01)

(52) **U.S. Cl.**
USPC **705/51**; 705/50; 713/176; 713/178; 726/22

(58) **Field of Classification Search** 705/1, 50-59
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

6,065,021	A *	5/2000	George	715/245
6,092,194	A *	7/2000	Touboul	726/24
6,421,709	B1 *	7/2002	McCormick et al.	709/206
6,697,950	B1 *	2/2004	Ko	726/24
6,792,536	B1 *	9/2004	Teppler	713/178
6,910,077	B2 *	6/2005	Najork	709/229
7,100,049	B2 *	8/2006	Gasparini et al.	713/170
7,114,147	B2 *	9/2006	Ballantyne et al.	717/107
2002/0112008	A1 *	8/2002	Christenson et al.	709/206
2004/0078422	A1 *	4/2004	Toomey	709/202
2004/0088433	A1 *	5/2004	Kaler et al.	709/246
2004/0111673	A1 *	6/2004	Bowman et al.	715/513
2005/0044242	A1 *	2/2005	Stevens et al.	709/228
2005/0154601	A1 *	7/2005	Halpern et al.	705/1
2007/0101423	A1 *	5/2007	Oliver et al.	726/22

* cited by examiner

Primary Examiner — Calvin L Hewitt, II
Assistant Examiner — C. Aaron McIntyre

(57) **ABSTRACT**

In some embodiments, techniques for computer security comprise displaying an electronic document, detecting a request to traverse a link, such as a hyperlink or a form submission, wherein the link is associated with an element of the document, evaluating an attribute, wherein the attribute is associated with the element of the document, and determining whether to perform the action based on the evaluation. Applications of these techniques include mitigating the effect of an attempt to modify web pages for fraudulent purposes, such as by a "phishing" attack incorporating malicious scripting.

19 Claims, 7 Drawing Sheets

Patent of Interest:

US8423471
(Priority date: Feb 04, 2004)

Protected document elements

Exemplary Market Applications:

The patented technology finds applications in web security.

US 8423471 – Claim 8

8. A non-transitory computer readable medium storing computer instructions which when executed by a processor cause the processor to perform the steps of:

   displaying an electronic document;

   detecting a request to traverse a link, wherein the link is associated with an element of the document;

   evaluating an attribute, wherein the attribute is associated with the element of the document and wherein evaluating the attribute includes determining whether a destination associated with the link was enumerated; and

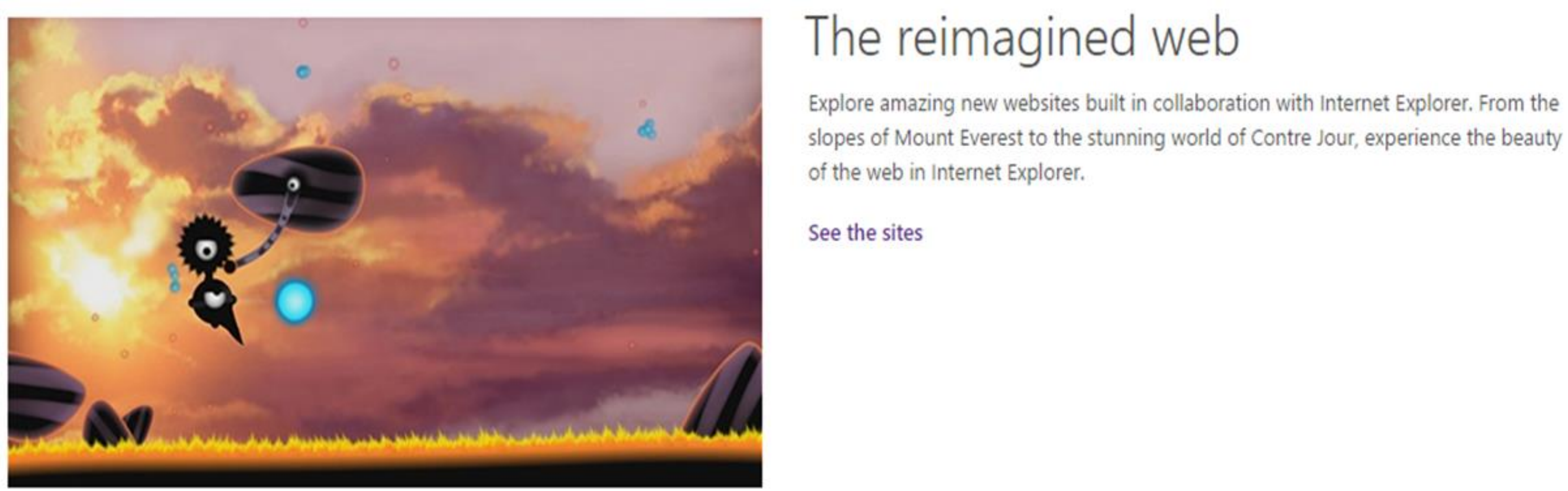
   determining whether to traverse the link based on the evaluation.

US 8423471 – EoU Summary

Microsoft Internet Explorer	
Key claim(s)	1, 7, and 8 (independent claims)
Mapped product	Chart has been made with respect to Internet Explorer. http://windows.microsoft.com/en-IN/internet-explorer/download-ie
Source	Information related to product is available at: http://windows.microsoft.com/en-IN/internet-explorer/download-ie http://windows.microsoft.com/en-IN/internet-explorer/products/ie-9/features/smartscreen-filter http://windows.microsoft.com/en-in/windows-vista/phishing-filter-frequently-asked-questions http://windows.microsoft.com/en-in/windows7/smartscreen-filter-frequently-asked-questions-ie9
Product launch date	2006 (Internet Explorer 7)
Details of product	Internet Explorer is a web browser by Microsoft. The browser includes SmartScreen Filter option which prevents the computer from accessing the websites that contain malwares. Also, it protects users from phishing.

US 8423471 – Microsoft Internet Explorer: Overview

Internet Explorer is a web browser by Microsoft. The browser includes SmartScreen Filter option which prevents the computer from accessing the websites that contain malwares. Also, it protects users from phishing.



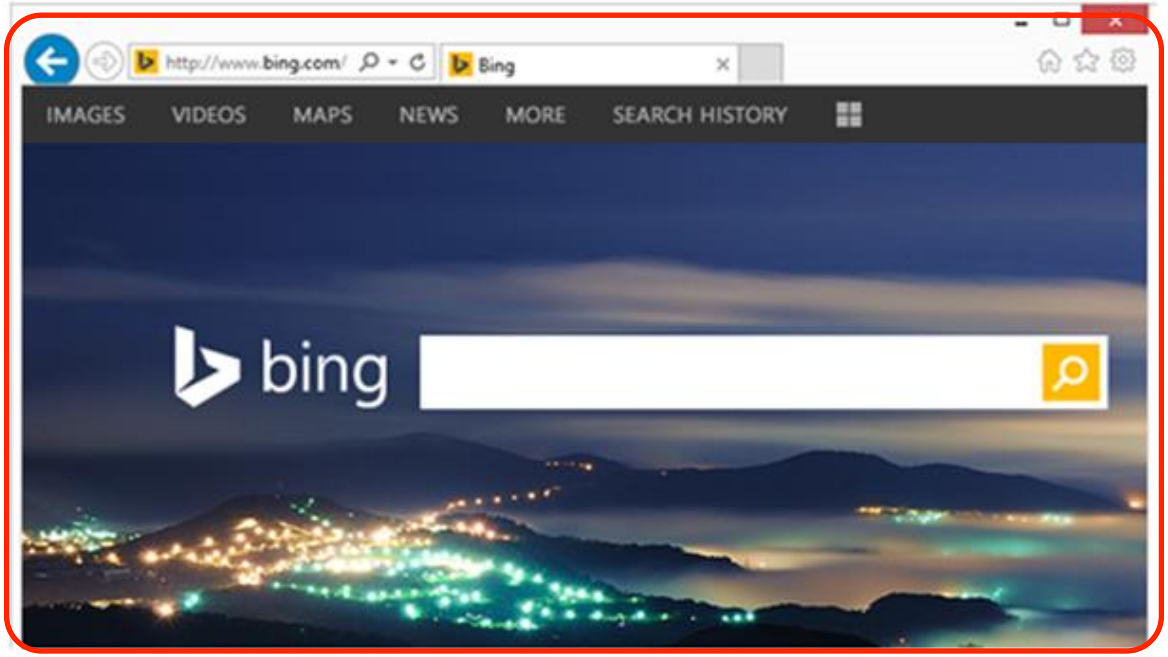
The reimagined web

Explore amazing new websites built in collaboration with Internet Explorer. From the slopes of Mount Everest to the stunning world of Contre Jour, experience the beauty of the web in Internet Explorer.

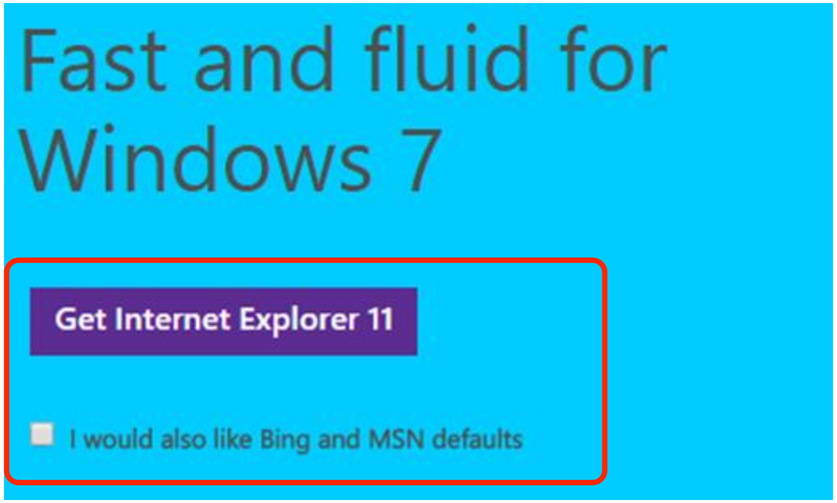
See the sites

Source: http://windows.microsoft.com/en-IN/internet-explorer/download-ie

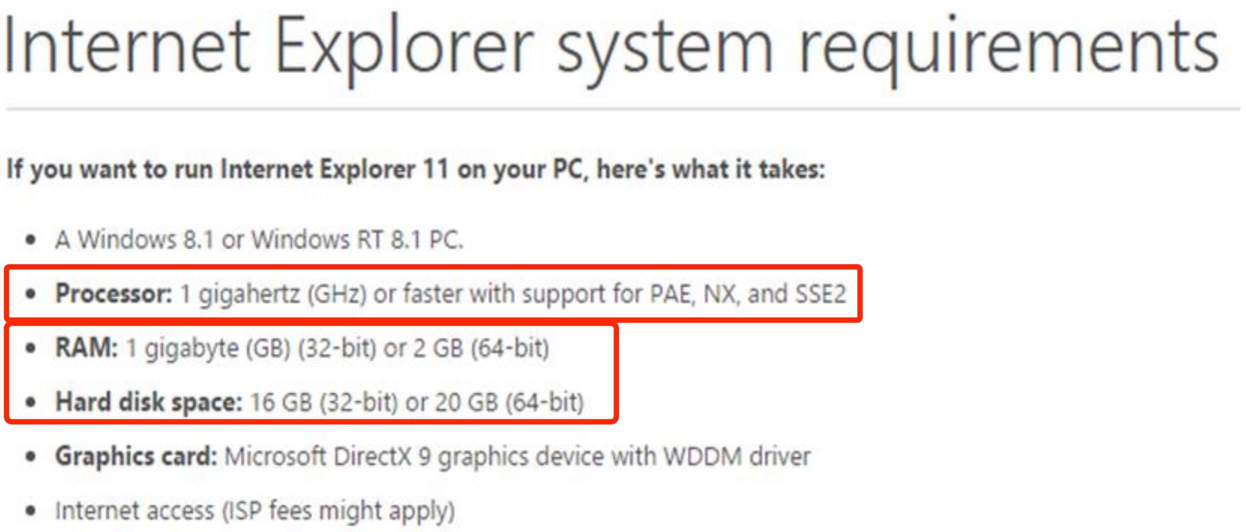
US 8423471 – Claim 8 vs. Microsoft Internet Explorer

Claim	Microsoft Internet Explorer
8. **A non-transitory computer readable medium storing computer instructions which when executed by a processor cause the processor to perform the steps of:** displaying an electronic document; detecting a request to traverse a link, wherein the link is associated with an element of the document; evaluating an attribute, wherein the attribute is associated with the element of the document and wherein evaluating the attribute includes determining whether a destination associated with the link was enumerated; and determining whether to traverse the link based on the evaluation.	As shown in the snapshot below, Internet Explorer is installed in the personal computers. The computers on which the Internet Explorer is installed includes a non-transitory computer readable medium (RAM).  http://www.bing.com/ Bing IMAGES VIDEOS MAPS NEWS MORE SEARCH HISTORY bing Source: http://windows.microsoft.com/en-IN/internet-explorer/download-ie

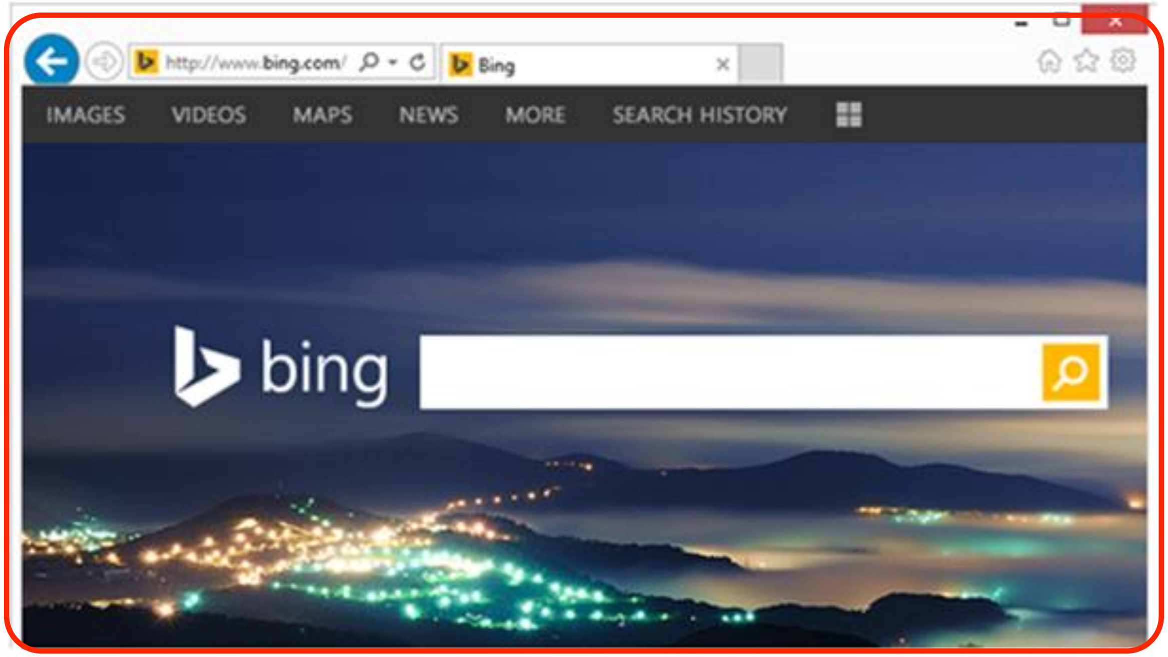
US 8423471 – Claim 8 vs. Microsoft Internet Explorer

Claim	Microsoft Internet Explorer
8. **A non-transitory computer readable medium storing computer instructions which when executed by a processor cause the processor to perform the steps of:** displaying an electronic document; detecting a request to traverse a link, wherein the link is associated with an element of the document; evaluating an attribute, wherein the attribute is associated with the element of the document and wherein evaluating the attribute includes determining whether a destination associated with the link was enumerated; and determining whether to traverse the link based on the evaluation.	As shown in the snapshot below, Internet Explorer software includes the computer instructions. Further, the computer instructions are executed by the processor. The processor is included in the computer in which Internet Explorer is installed.  Fast and fluid for Windows 7 Get Internet Explorer 11 ☐ I would also like Bing and MSN defaults Source: http://windows.microsoft.com/en-IN/internet-explorer/download-ie

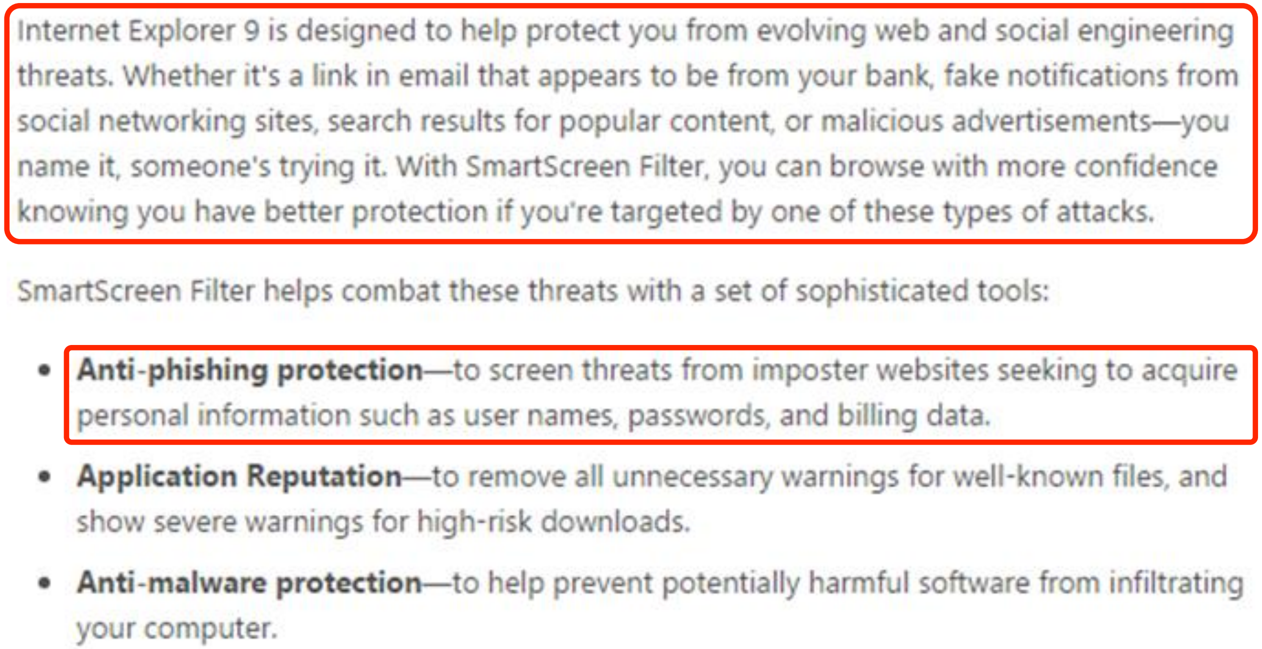
US 8423471 – Claim 8 vs. Microsoft Internet Explorer

Claim	Microsoft Internet Explorer
8. **A non-transitory computer readable medium storing computer instructions which when executed by a processor cause the processor to perform the steps of:** displaying an electronic document; detecting a request to traverse a link, wherein the link is associated with an element of the document; evaluating an attribute, wherein the attribute is associated with the element of the document and wherein evaluating the attribute includes determining whether a destination associated with the link was enumerated; and determining whether to traverse the link based on the evaluation.	As shown in the snapshot below, Internet Explorer has specific system requirements of free disk space and processor. Further, Internet Explorer includes computer instructions which are executed by a processor of the personal computer.  Internet Explorer system requirements **If you want to run Internet Explorer 11 on your PC, here's what it takes:** • A Windows 8.1 or Windows RT 8.1 PC. • **Processor:** 1 gigahertz (GHz) or faster with support for PAE, NX, and SSE2 • **RAM:** 1 gigabyte (GB) (32-bit) or 2 GB (64-bit) • **Hard disk space:** 16 GB (32-bit) or 20 GB (64-bit) • **Graphics card:** Microsoft DirectX 9 graphics device with WDDM driver • Internet access (ISP fees might apply) Source: http://windows.microsoft.com/en-in/internet-explorer/ie-system-requirements#ie=ie-11

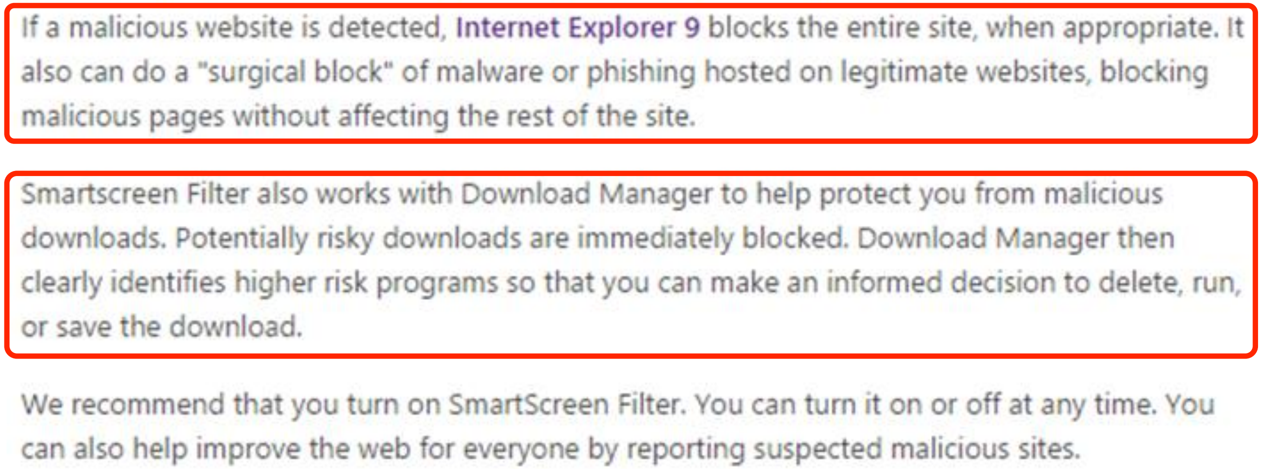
US 8423471 – Claim 8 vs. Microsoft Internet Explorer

Claim	Microsoft Internet Explorer
8. A non-transitory computer readable medium storing computer instructions which when executed by a processor cause the processor to perform the steps of: **displaying an electronic document;** detecting a request to traverse a link, wherein the link is associated with an element of the document; evaluating an attribute, wherein the attribute is associated with the element of the document and wherein evaluating the attribute includes determining whether a destination associated with the link was enumerated; and determining whether to traverse the link based on the evaluation.	As shown in the snapshot below, Internet Explorer displays an electronic document. The electronic document is the HTML/web page.  http://www.bing.com/ Bing IMAGES VIDEOS MAPS NEWS MORE SEARCH HISTORY bing Source: http://windows.microsoft.com/en-IN/internet-explorer/download-ie

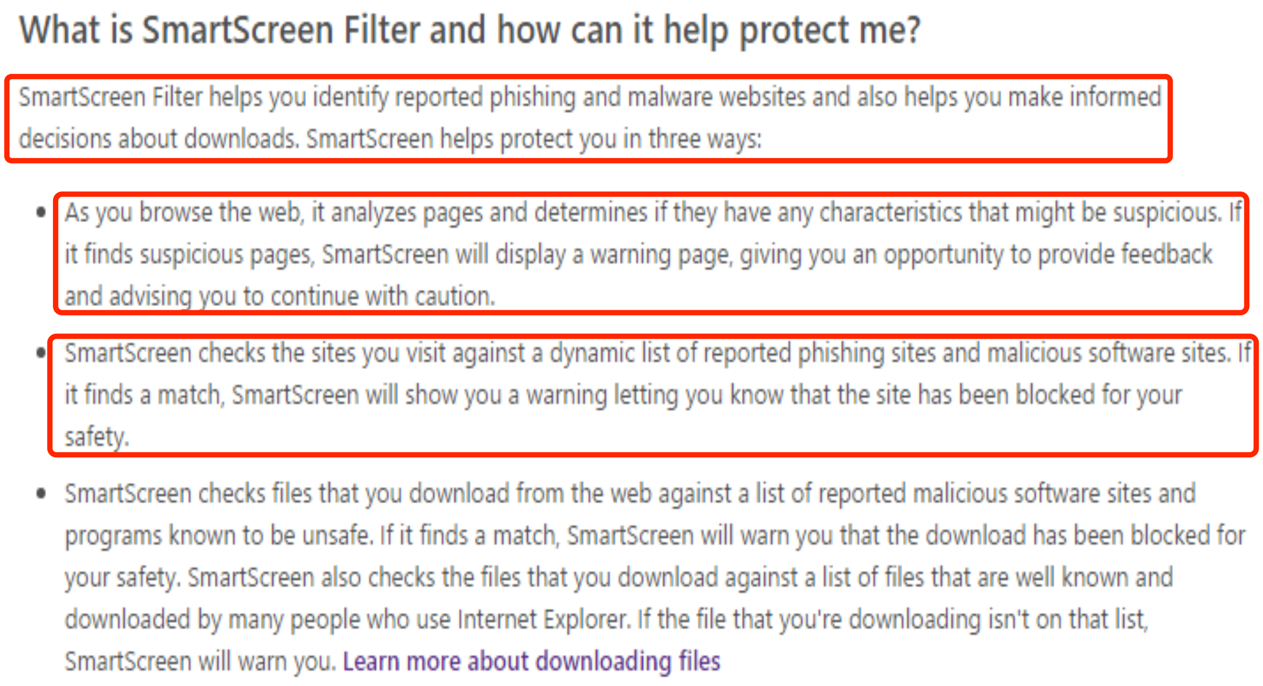
US 8423471 – Claim 8 vs. Microsoft Internet Explorer

Claim	Microsoft Internet Explorer
8. A non-transitory computer readable medium storing computer instructions which when executed by a processor cause the processor to perform the steps of: displaying an electronic document; **detecting a request to traverse a link, wherein the link is associated with an element of the document;** evaluating an attribute, wherein the attribute is associated with the element of the document and wherein evaluating the attribute includes determining whether a destination associated with the link was enumerated; and determining whether to traverse the link based on the evaluation.	As shown in the snapshot below, Internet Explorer detects a request to traverse a hyperlink included in the web page. The request is made by the user when the user clicks the hyperlink. As the user clicks the hyperlink, phishing and malware alerts are generated if the website is suspicious. Further, the hyperlink included in the web page is a part of the electronic document (web page). Thus, Internet Explorer detects a request to traverse a link, wherein the link is associated with the element of the document.  Internet Explorer 9 is designed to help protect you from evolving web and social engineering threats. Whether it's a link in email that appears to be from your bank, fake notifications from social networking sites, search results for popular content, or malicious advertisements—you name it, someone's trying it. With SmartScreen Filter, you can browse with more confidence knowing you have better protection if you're targeted by one of these types of attacks. SmartScreen Filter helps combat these threats with a set of sophisticated tools: - **Anti-phishing protection**—to screen threats from imposter websites seeking to acquire personal information such as user names, passwords, and billing data. - **Application Reputation**—to remove all unnecessary warnings for well-known files, and show severe warnings for high-risk downloads. - **Anti-malware protection**—to help prevent potentially harmful software from infiltrating your computer. Source: http://windows.microsoft.com/en-IN/internet-explorer/products/ie-9/features/smartscreen-filter

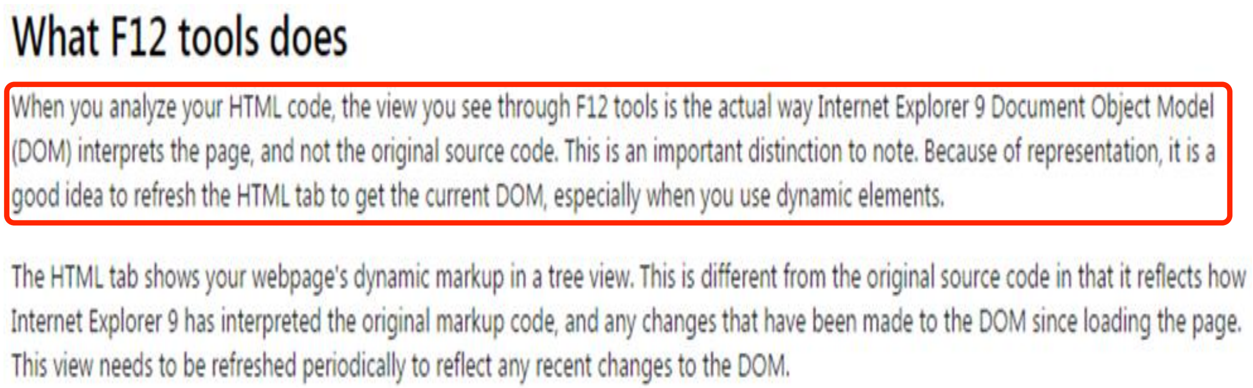
US 8423471 – Claim 8 vs. Microsoft Internet Explorer

Claim	Microsoft Internet Explorer
8. A non-transitory computer readable medium storing computer instructions which when executed by a processor cause the processor to perform the steps of: displaying an electronic document; **detecting a request to traverse a link, wherein the link is associated with an element of the document;** evaluating an attribute, wherein the attribute is associated with the element of the document and wherein evaluating the attribute includes determining whether a destination associated with the link was enumerated; and determining whether to traverse the link based on the evaluation.	As shown in the snapshot below, Internet Explorer provides warning when the browser attempts to access the suspicious web sites. This request is detected by the Internet Explorer browser. The request includes a link that needs to be accessed by the user.  If a malicious website is detected, Internet Explorer 9 blocks the entire site, when appropriate. It also can do a "surgical block" of malware or phishing hosted on legitimate websites, blocking malicious pages without affecting the rest of the site. Smartscreen Filter also works with Download Manager to help protect you from malicious downloads. Potentially risky downloads are immediately blocked. Download Manager then clearly identifies higher risk programs so that you can make an informed decision to delete, run, or save the download. We recommend that you turn on SmartScreen Filter. You can turn it on or off at any time. You can also help improve the web for everyone by reporting suspected malicious sites. Source: http://windows.microsoft.com/en-IN/internet-explorer/products/ie-9/features/smartscreen-filter

US 8423471 – Claim 8 vs. Microsoft Internet Explorer

Claim	Microsoft Internet Explorer
8. A non-transitory computer readable medium storing computer instructions which when executed by a processor cause the processor to perform the steps of: displaying an electronic document; **detecting a request to traverse a link, wherein the link is associated with an element of the document;** evaluating an attribute, wherein the attribute is associated with the element of the document and wherein evaluating the attribute includes determining whether a destination associated with the link was enumerated; and determining whether to traverse the link based on the evaluation.	As shown in the snapshot below, Internet Explorer includes SmartScreen Filter that provides warning when the browser attempts to access the suspicious web sites. This request is detected by the Internet Explorer browser.  What is SmartScreen Filter and how can it help protect me? SmartScreen Filter helps you identify reported phishing and malware websites and also helps you make informed decisions about downloads. SmartScreen helps protect you in three ways: • As you browse the web, it analyzes pages and determines if they have any characteristics that might be suspicious. If it finds suspicious pages, SmartScreen will display a warning page, giving you an opportunity to provide feedback and advising you to continue with caution. • SmartScreen checks the sites you visit against a dynamic list of reported phishing sites and malicious software sites. If it finds a match, SmartScreen will show you a warning letting you know that the site has been blocked for your safety. • SmartScreen checks files that you download from the web against a list of reported malicious software sites and programs known to be unsafe. If it finds a match, SmartScreen will warn you that the download has been blocked for your safety. SmartScreen also checks the files that you download against a list of files that are well known and downloaded by many people who use Internet Explorer. If the file that you're downloading isn't on that list, SmartScreen will warn you. Learn more about downloading files Source: http://windows.microsoft.com/en-in/internet-explorer/use-smartscreen-filter#ie=ie-11

US 8423471 – Claim 8 vs. Microsoft Internet Explorer

Claim	Microsoft Internet Explorer
8. A non-transitory computer readable medium storing computer instructions which when executed by a processor cause the processor to perform the steps of: displaying an electronic document; **detecting a request to traverse a link, wherein the link is associated with an element of the document;** evaluating an attribute, wherein the attribute is associated with the element of the document and wherein evaluating the attribute includes determining whether a destination associated with the link was enumerated; and determining whether to traverse the link based on the evaluation.	As shown in the snapshot below, web pages include elements. The hyperlink included in the webpage (electronic document) is associated with the element of the document object model.  What F12 tools does When you analyze your HTML code, the view you see through F12 tools is the actual way Internet Explorer 9 Document Object Model (DOM) interprets the page, and not the original source code. This is an important distinction to note. Because of representation, it is a good idea to refresh the HTML tab to get the current DOM, especially when you use dynamic elements. The HTML tab shows your webpage's dynamic markup in a tree view. This is different from the original source code in that it reflects how Internet Explorer 9 has interpreted the original markup code, and any changes that have been made to the DOM since loading the page. This view needs to be refreshed periodically to reflect any recent changes to the DOM. Source: https://msdn.microsoft.com/en-us/library/gg589512(v=vs.85).aspx

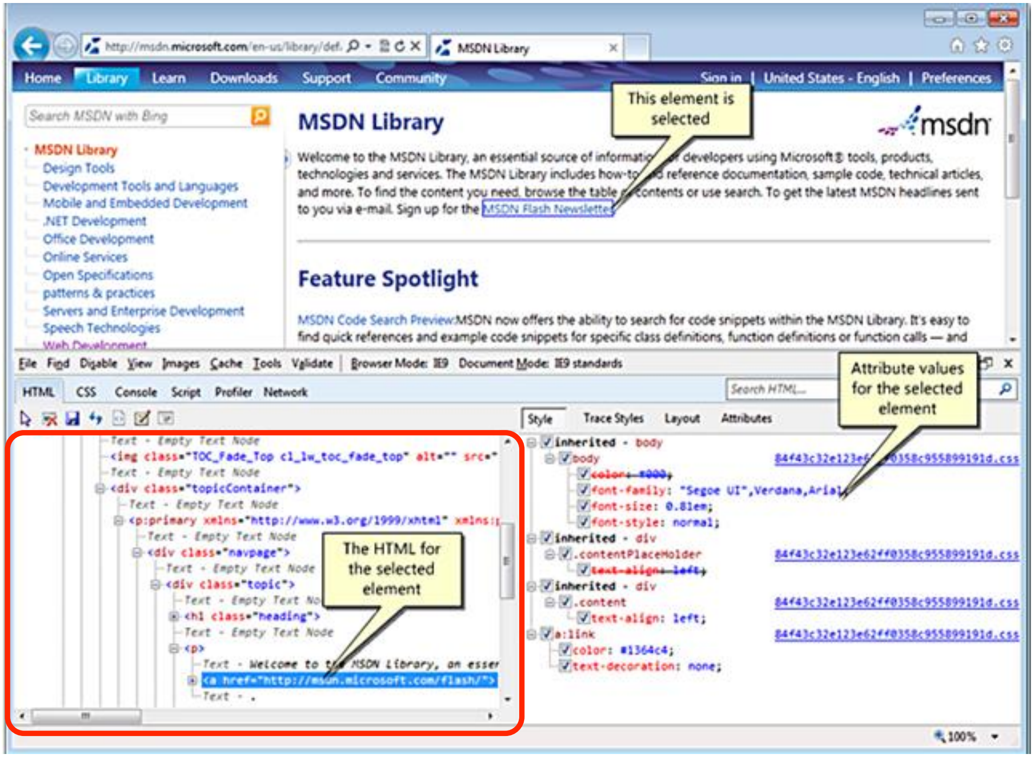
US 8423471 – Claim 8 vs. Microsoft Internet Explorer

Claim

8. A non-transitory computer readable medium storing computer instructions which when executed by a processor cause the processor to perform the steps of: displaying an electronic document; **detecting a request to traverse a link, wherein the link is associated with an element of the document;** evaluating an attribute, wherein the attribute is associated with the element of the document and wherein evaluating the attribute includes determining whether a destination associated with the link was enumerated; and determining whether to traverse the link based on the evaluation.

Microsoft Internet Explorer

As shown in the snapshot below, web pages include elements. The hyperlink included in the webpage (electronic document) is associated with the element of the document object model.



http://msdn.microsoft.com/en-us/library/def.
MSDN Library
Home
Library
Learn
Downloads
Support
Community
Sign in | United States - English | Preferences
Search MSDN with Bing
MSDN Library
Design Tools
Development Tools and Languages
Mobile and Embedded Development
.NET Development
Office Development
Online Services
Open Specifications
patterns & practices
Servers and Enterprise Development
Speech Technologies
Web Development
msdn
File Find Disable View Images Cache Tools Validate Browser Mode: IE9 Document Mode: IE9 standards
HTML CSS Console Script Profiler Network
Search HTML...
Style Trace Styles Layout Attributes
100%

MSDN Library

Welcome to the MSDN Library, an essential source of information for developers using Microsoft® tools, products, technologies and services. The MSDN Library includes how-to and reference documentation, sample code, technical articles, and more. To find the content you need, browse the table of contents or use search. To get the latest MSDN headlines sent to you via e-mail. Sign up for the MSDN Flash Newsletter

Feature Spotlight

MSDN Code Search Preview:MSDN now offers the ability to search for code snippets within the MSDN Library. It's easy to find quick references and example code snippets for specific class definitions, function definitions or function calls — and

```
Text - Empty Text Node
<img class="TOC_Fade_Top cl_lw_toc_fade_top" alt="" src="
Text - Empty Text Node
<div class="topicContainer">
  Text - Empty Text Node
  <p:primary xmlns="http://www.w3.org/1999/xhtml" xmlns:p
    Text - Empty Text Node
    <div class="navpage">
      Text - Empty Text Node
      <div class="topic">
        Text - Empty Text No
        <h1 class="heading">
        Text - Empty Text Node
        <p>
          Text - Welcome to the MSDN Library, an esser
          <a href="http://msdn.microsoft.com/flash/">
          Text - .
```

```
inherited - body
  body                                84f43c32e123e62ff0358c955899191d.css
    ~~color: #000;~~
    font-family: "Segoe UI",Verdana,Arial;
    font-size: 0.81em;
    font-style: normal;
inherited - div
  .contentPlaceHolder                 84f43c32e123e62ff0358c955899191d.css
    ~~text-align: left;~~
inherited - div
  .content                            84f43c32e123e62ff0358c955899191d.css
    text-align: left;
a:link                                84f43c32e123e62ff0358c955899191d.css
  color: #1364c4;
  text-decoration: none;
```

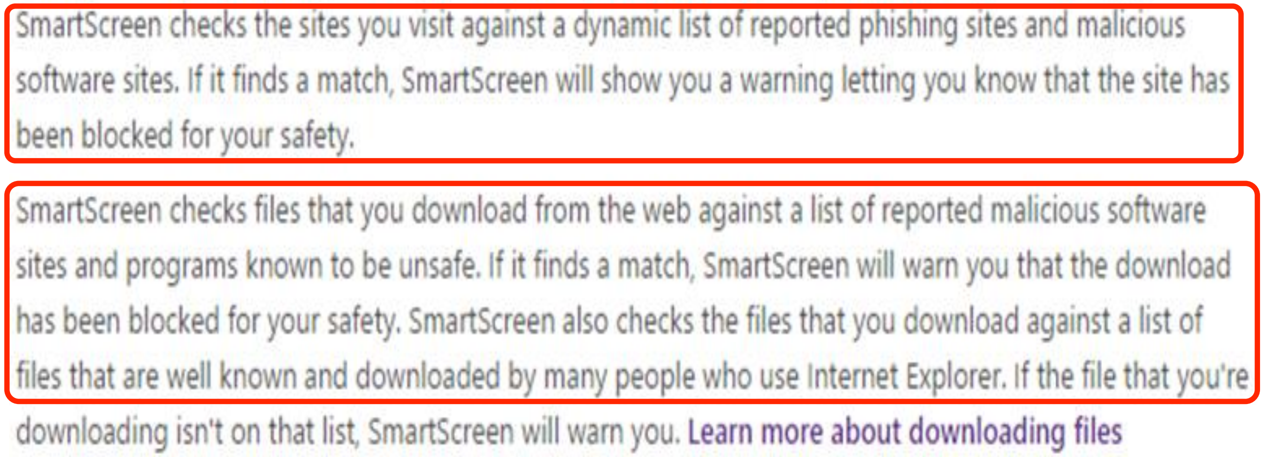
This element is selected

Attribute values for the selected element

The HTML for the selected element

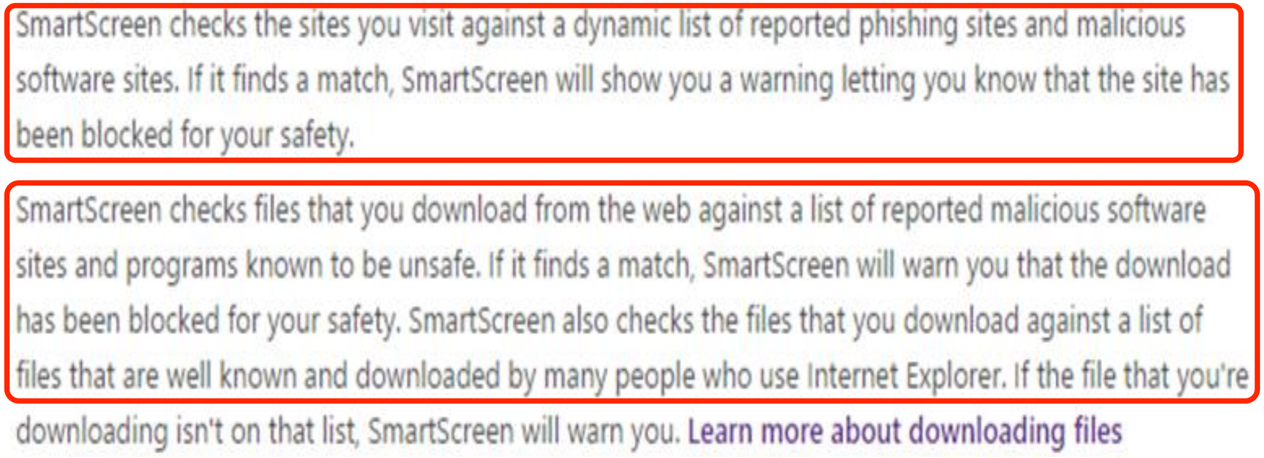
Source: https://msdn.microsoft.com/en-us/library/gg589512(v=vs.85).aspx

US 8423471 – Claim 8 vs. Microsoft Internet Explorer

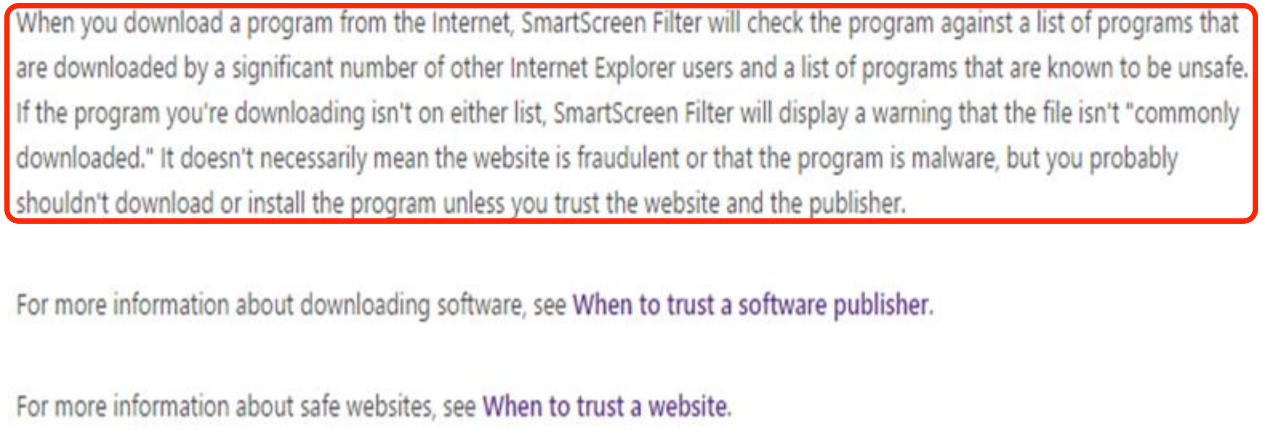
Claim	Microsoft Internet Explorer
8. A non-transitory computer readable medium storing computer instructions which when executed by a processor cause the processor to perform the steps of: displaying an electronic document; detecting a request to traverse a link, wherein the link is associated with an element of the document; **evaluating an attribute, wherein the attribute is associated with the element of the document** and wherein evaluating the attribute includes determining whether a destination associated with the link was enumerated; and determining whether to traverse the link based on the evaluation.	As shown in the snapshot below, Internet Explorer evaluates the safety attribute (malware or phishing websites) related to requested link. The link is associated with the element of the document. Thus, Internet Explorer evaluates safety attributes associated with the element of the document.  SmartScreen checks the sites you visit against a dynamic list of reported phishing sites and malicious software sites. If it finds a match, SmartScreen will show you a warning letting you know that the site has been blocked for your safety. SmartScreen checks files that you download from the web against a list of reported malicious software sites and programs known to be unsafe. If it finds a match, SmartScreen will warn you that the download has been blocked for your safety. SmartScreen also checks the files that you download against a list of files that are well known and downloaded by many people who use Internet Explorer. If the file that you're downloading isn't on that list, SmartScreen will warn you. Learn more about downloading files Source: http://windows.microsoft.com/en-in/internet-explorer/use-smartscreen-filter#ie=ie-11

© ICEBERG Capital Partners Limited 2015. Confidential

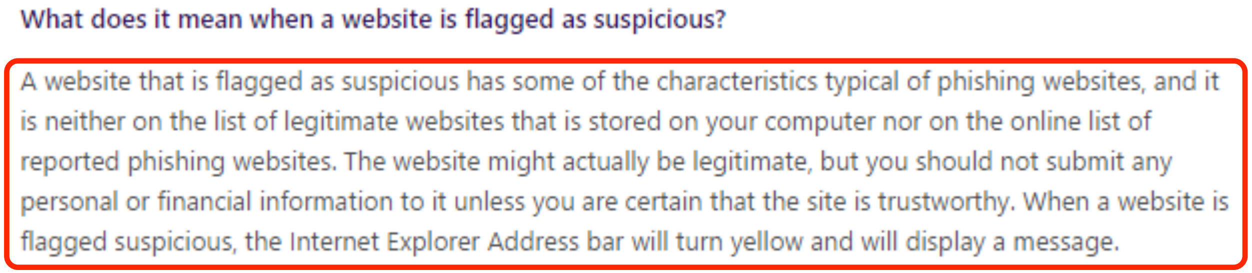
US 8423471 – Claim 8 vs. Microsoft Internet Explorer

Claim	Microsoft Internet Explorer
8. A non-transitory computer readable medium storing computer instructions which when executed by a processor cause the processor to perform the steps of: displaying an electronic document; detecting a request to traverse a link, wherein the link is associated with an element of the document; evaluating an attribute, wherein the attribute is associated with the element of the document **and wherein evaluating the attribute includes determining whether a destination associated with the link was enumerated;** and determining whether to traverse the link based on the evaluation.	As shown in the snapshot below, Internet Explorer evaluates the safety attribute of the hyperlink included in the web page. The SmartScreen Filter option of the Internet Explorer enables the checking of the URL against the list of suspected websites.  SmartScreen checks the sites you visit against a dynamic list of reported phishing sites and malicious software sites. If it finds a match, SmartScreen will show you a warning letting you know that the site has been blocked for your safety. SmartScreen checks files that you download from the web against a list of reported malicious software sites and programs known to be unsafe. If it finds a match, SmartScreen will warn you that the download has been blocked for your safety. SmartScreen also checks the files that you download against a list of files that are well known and downloaded by many people who use Internet Explorer. If the file that you're downloading isn't on that list, SmartScreen will warn you. Learn more about downloading files Source: http://windows.microsoft.com/en-in/internet-explorer/use-smartscreen-filter#ie=ie-11

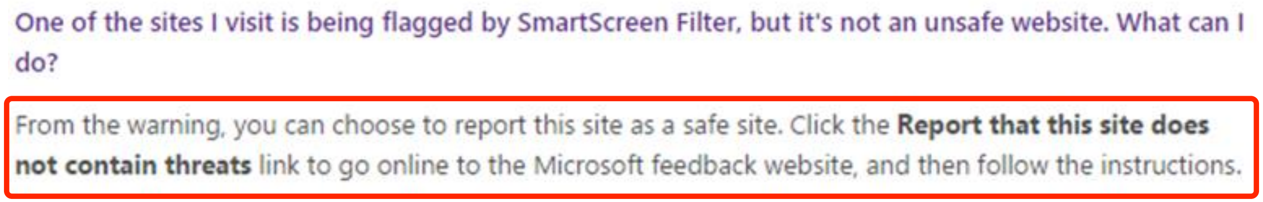
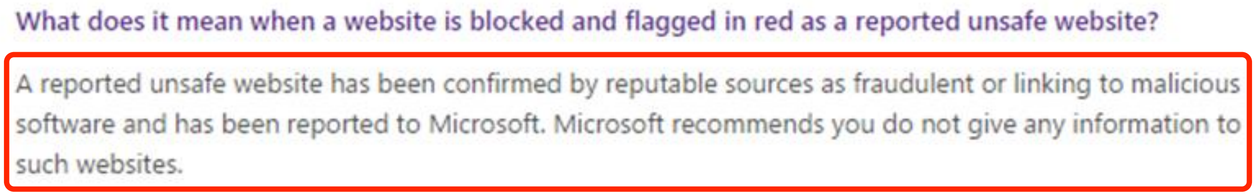
US 8423471 – Claim 8 vs. Microsoft Internet Explorer

Claim	Microsoft Internet Explorer
8. A non-transitory computer readable medium storing computer instructions which when executed by a processor cause the processor to perform the steps of: displaying an electronic document; detecting a request to traverse a link, wherein the link is associated with an element of the document; evaluating an attribute, wherein the attribute is associated with the element of the document **and wherein evaluating the attribute includes determining whether a destination associated with the link was enumerated;** and determining whether to traverse the link based on the evaluation.	As shown in the snapshot below, Internet Explorer evaluates the safety attribute of the hyperlink included in the web page. The SmartScreen Filter option of the Internet Explorer enables the checking of the URL against the list of suspected websites. The enumerated websites are the destinations associated with the hyperlink included in the document.  When you download a program from the Internet, SmartScreen Filter will check the program against a list of programs that are downloaded by a significant number of other Internet Explorer users and a list of programs that are known to be unsafe. If the program you're downloading isn't on either list, SmartScreen Filter will display a warning that the file isn't "commonly downloaded." It doesn't necessarily mean the website is fraudulent or that the program is malware, but you probably shouldn't download or install the program unless you trust the website and the publisher. For more information about downloading software, see When to trust a software publisher. For more information about safe websites, see When to trust a website. Source: http://windows.microsoft.com/en-in/windows7/smartscreen-filter-frequently-asked-questions-ie9

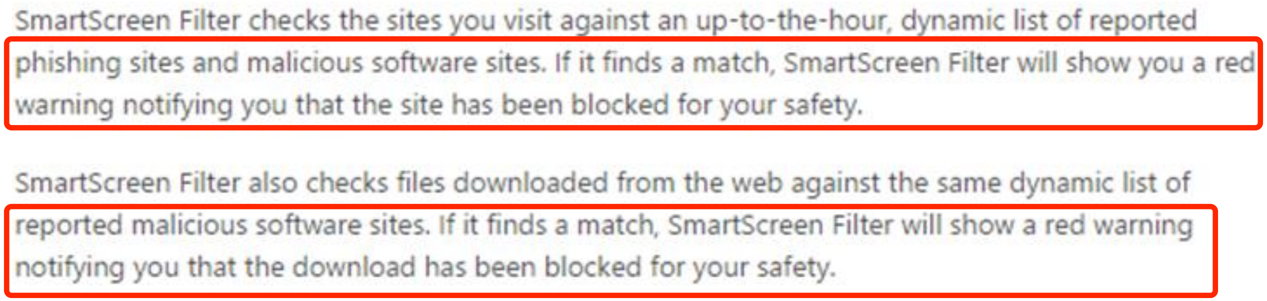
US 8423471 – Claim 8 vs. Microsoft Internet Explorer

Claim	Microsoft Internet Explorer
8. A non-transitory computer readable medium storing computer instructions which when executed by a processor cause the processor to perform the steps of: displaying an electronic document; detecting a request to traverse a link, wherein the link is associated with an element of the document; evaluating an attribute, wherein the attribute is associated with the element of the document **and wherein evaluating the attribute includes determining whether a destination associated with the link was enumerated;** and determining whether to traverse the link based on the evaluation.	As shown in the snapshot below, Internet Explorer evaluates the safety attribute of the hyperlink included in the web page. The SmartScreen Filter option of the Internet Explorer enables the checking of the URL against the list of suspected websites. The enumerated websites are the destinations associated with the hyperlink included in the document. Thus, Internet Explorer evaluates the safety attribute by determining whether the destination (URL/web page) associated with the hyperlink was enumerated.  **What does it mean when a website is flagged as suspicious?** A website that is flagged as suspicious has some of the characteristics typical of phishing websites, and it is neither on the list of legitimate websites that is stored on your computer nor on the online list of reported phishing websites. The website might actually be legitimate, but you should not submit any personal or financial information to it unless you are certain that the site is trustworthy. When a website is flagged suspicious, the Internet Explorer Address bar will turn yellow and will display a message. Source: http://windows.microsoft.com/en-in/windows-vista/phishing-filter-frequently-asked-questions

US 8423471 – Claim 8 vs. Microsoft Internet Explorer

Claim	Microsoft Internet Explorer
8. A non-transitory computer readable medium storing computer instructions which when executed by a processor cause the processor to perform the steps of: displaying an electronic document; detecting a request to traverse a link, wherein the link is associated with an element of the document; evaluating an attribute, wherein the attribute is associated with the element of the document and wherein evaluating the attribute includes determining whether a destination associated with the link was enumerated; **and determining whether to traverse the link based on the evaluation.**	As shown in the snapshot below, Internet Explorer determines whether to traverse the link based on the evaluation. Internet Explorer allows the user to traverse the link if the evaluation states that the website is safe to visit. If the link is included in the malware, phishing or unwanted software list, Internet Explorer provides a warning before traversing the link. Thus, Internet Explorer determines whether to traverse the link based on the evaluation.  One of the sites I visit is being flagged by SmartScreen Filter, but it's not an unsafe website. What can I do? From the warning, you can choose to report this site as a safe site. Click the **Report that this site does not contain threats** link to go online to the Microsoft feedback website, and then follow the instructions.  What does it mean when a website is blocked and flagged in red as a reported unsafe website? A reported unsafe website has been confirmed by reputable sources as fraudulent or linking to malicious software and has been reported to Microsoft. Microsoft recommends you do not give any information to such websites. Source: http://windows.microsoft.com/en-in/windows/smartscreen-filter-faq#1TC=windows-7

US 8423471 – Claim 8 vs. Microsoft Internet Explorer

Claim	Microsoft Internet Explorer
8. A non-transitory computer readable medium storing computer instructions which when executed by a processor cause the processor to perform the steps of: displaying an electronic document; detecting a request to traverse a link, wherein the link is associated with an element of the document; evaluating an attribute, wherein the attribute is associated with the element of the document and wherein evaluating the attribute includes determining whether a destination associated with the link was enumerated; **and determining whether to traverse the link based on the evaluation.**	As shown in the snapshot below, Internet Explorer determines whether to traverse the link based on the evaluation. Internet Explorer allows the user to traverse the link if the evaluation states that the website is safe to visit. If the link is included in the malware, phishing or unwanted software list, Internet Explorer provides a warning before traversing the link. Thus, Internet Explorer determines whether to traverse the link based on the evaluation.  SmartScreen Filter checks the sites you visit against an up-to-the-hour, dynamic list of reported phishing sites and malicious software sites. If it finds a match, SmartScreen Filter will show you a red warning notifying you that the site has been blocked for your safety. SmartScreen Filter also checks files downloaded from the web against the same dynamic list of reported malicious software sites. If it finds a match, SmartScreen Filter will show a red warning notifying you that the download has been blocked for your safety. Source: http://windows.microsoft.com/en-in/windows/smartscreen-filter-faq#1TC=windows-7

If the portfolio is of interest or you require further information, please contact your ICEBERG relationship manager.

ICEBERG Capital Partners Limited
35 Berkeley Square
Mayfair, London
England
W1J 5BF
UK

P. +44 (0)207 887 6377
F. +44 (0)207 681 2137
E. enquiries@iceberg-cap.com
W. www.iceberg-cap.com

Disclaimer and Notice:

The information in this document is provided in confidence for the sole purpose of supporting the independent evaluation of the enclosed patent portfolio by potential buyers. Any discussion of the use or potential use of the patent portfolio is for illustrative purposes only. This document, the offer of the portfolio for sale, and any materials or information exchanged during the sales process — whether in this document or otherwise — are not, are not intended to be, and should not be construed as being, notice of infringement, any form of accusation of infringement, or any opinion regarding the actual use of the patent portfolio.

No assurances, representations or warranties pertaining to the patent portfolio or its validity are provided or implied herein, and the information in this document is not legal advice, analysis or a legal opinion. Potential purchasers must rely on their own evaluation, examination and due diligence of the patent portfolio, as this document is solely attributable to ICEBERG Capital Partners Limited and does not necessarily represent the views or opinions of the seller.

This document and any other materials or information provided by ICEBERG Capital Partners Limited related to the portfolio are copyrighted, and are intended for use by the receiving party solely for its use in participating in the sales process and in determining whether to purchase the portfolio. Any distribution of such materials or information outside of the receiving party's organisation without ICEBERG Capital Partners Limited permission is strictly prohibited.