goto; conference aarhus

Web Platform: The Secure Parts

Mike West
@mikewest, mkw.st/+

Slides: https://mkw.st/r/goto13

goto; conference aarhus

Web Platform: The Not Completely and Obviously Insecure Parts

Mike West
@mikewest, mkw.st/+

Slides: https://mkw.st/r/goto13

INTERNATIONAL SOFTWARE DEVELOPMENT CONFERENCE

gotocon.com

Malware Ahead!

ianfette.org

chrome

The Website Ahead Contains Malware!

Google Chrome has blocked access to ianfette.org for now.

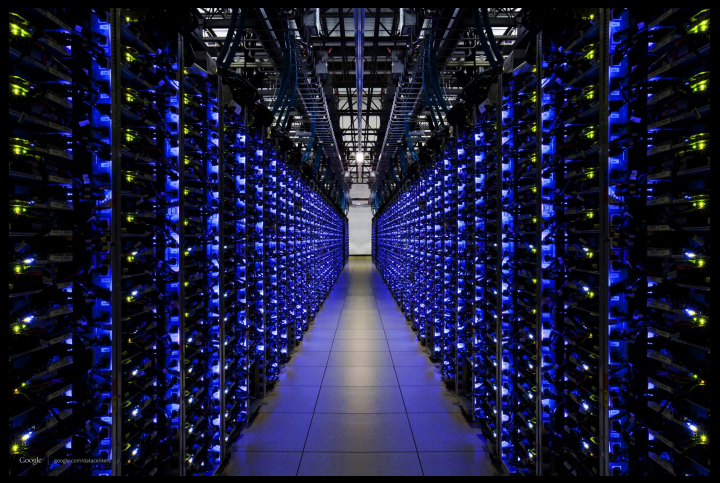
Even if you have visited this website safely in the past, visiting it now is very likely to infect your computer with malware.

Malware is malicious software that causes things like identity theft, financial loss, and permanent file deletion.

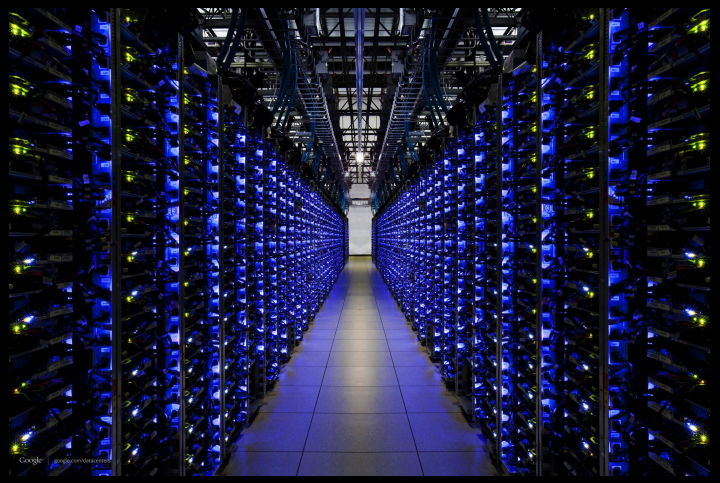
Learn more

Go back

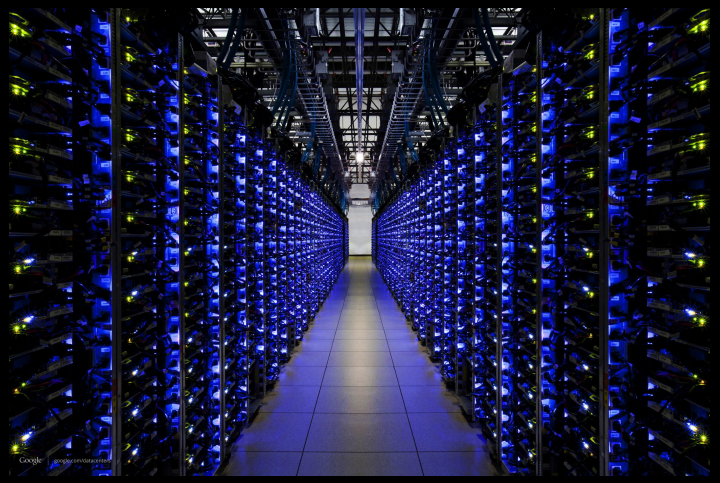
Advanced




chrome



goto; conference



goto; conference

StartSSL™ Certificates &

StartCom Ltd. (Start Commercial Limited) [IL] https://www.startssl.com

Sign-up For Free

StartSSL™ PKI

StartCom Home | StartSSL PKI | StartSSL DNS | StartSSL WoT | F. A. Q. | Control Panel

StartSSL™ - The Swiss Officer's Knife of Digital Certificates & PKI

- StartSSL™ Home
- StartSSL™ Products
- Comparison Chart
- How to Enroll
- How to Install
- HTML Examples
- F. A. Q.
- Policies & Resources
- Contact StartCom
- Control Panel

Welcome to StartSSL™ PKI

StartSSL™ is the trade mark of the **StartCom® Certification Authority** - a leader of the digital certification industry. We provide you with everything from **free** low-assurance **SSL** certificates up to the most advanced PKI and security solutions for your business and personal use.

StartSSL™ EV
High Level Trust Indicator
Time Limited Offer
Only US$ 199.90

SUPPORTED BROWSERS

PAYMENT OPTIONS

StartSSL™ (Class 1)
128/256-bit Encryption, **1 Year** Validity
Legitimate SSL/TLS + S.MIME Certificates
No Charge, Unlimited + 100 % Free

StartSSL™ Verified (Class 2)
128/256-bit Encryption, **2 Years** Validity
Legitimate SSL/TLS + S/MIME + Object Code
Wild Cards, Multiple Domain Names (UCC)
Unlimited Certificates - US$ 59.90

StartSSL™ Extended Validation
128/256-bit Encryption, **2 Years** Validity
Highest Level Third Party Assurance
Green Extended Trust Indicator
Multiple Domain Names (UCC)
Special Offer - US$ 199.90

High Protection
StartSSL™ High Level Protection
No MD5 Hashes, Weak Key Scans
Minimum 2048-bit Strong RSA Keys

Hardware
Aladdin® USB eToken Pro
Aladdin® Smart Cards + Reader
Original Driver Software + PKI Client
Enterprise PKI Customized Solutions

Authentication
StartSSL™ Authentication SSL Protected
Open Identity Authentication Provider
Click here to log into your
StartSSL™ Account

Internationally Recognized
WebTrust for CAs + WebTrust EV Certified
Recognized by major browsers + software vendors

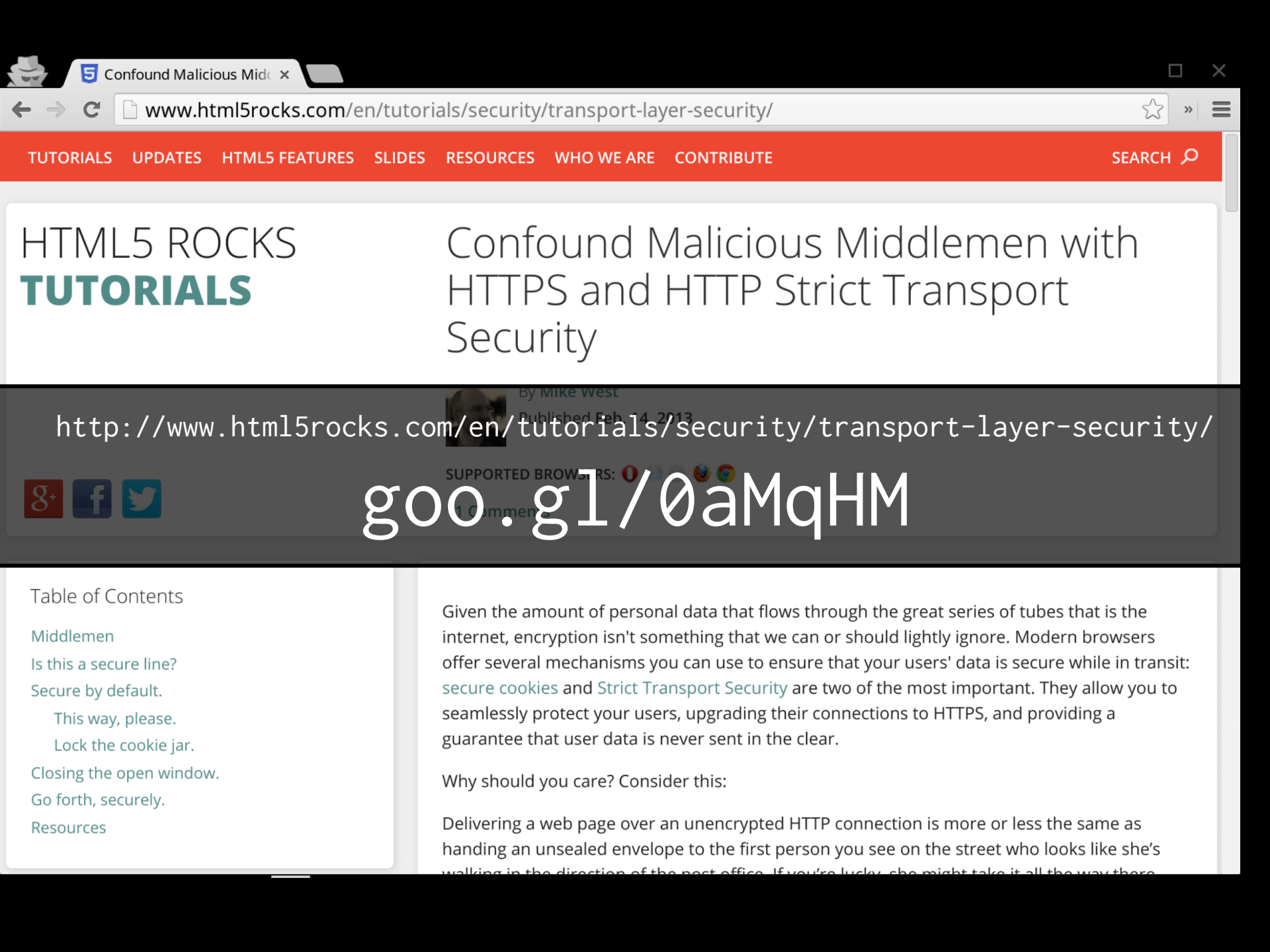
Easy Enrollment
Sign-up and you will receive right away an S/MIME client-certificate and a digital StartSSL™ Open Identity without charge during the easy three-step enrollment!

```
$ curl -I http://mkw.st/
HTTP/1.1 301 Moved Permanently
Server: nginx/1.5.0
Date: Mon, 1 Oct 2013 19:36:15 GMT
Content-Type: text/html
Content-Length: 184
Connection: keep-alive
Keep-Alive: timeout=20
Location: https://mkw.st/
```

```
$ curl -I https://mkw.st/
HTTP/1.1 200 OK
Server: nginx/1.5.0
Date: Mon, 1 Oct 2013 19:42:31 GMT
...
Strict-Transport-Security:
    max-age=2592000;
    includeSubDomains
...
```

```
Set-Cookie: ...; secure; HttpOnly
```

```
Public-Key-Pins:
    max-age=2592000;
    pin-sha1="4n972H…60yw4uqe/baXc="
```


Confound Malicious Mid
www.html5rocks.com/en/tutorials/security/transport-layer-security/
TUTORIALS UPDATES HTML5 FEATURES SLIDES RESOURCES WHO WE ARE CONTRIBUTE SEARCH
HTML5 ROCKS
TUTORIALS
Confound Malicious Middlemen with HTTPS and HTTP Strict Transport Security
By Mike West
Published Feb 14, 2013
SUPPORTED BROWSERS:
http://www.html5rocks.com/en/tutorials/security/transport-layer-security/
goo.gl/0aMqHM
Table of Contents
Middlemen
Is this a secure line?
Secure by default.
This way, please.
Lock the cookie jar.
Closing the open window.
Go forth, securely.
Resources
Given the amount of personal data that flows through the great series of tubes that is the internet, encryption isn't something that we can or should lightly ignore. Modern browsers offer several mechanisms you can use to ensure that your users' data is secure while in transit: secure cookies and Strict Transport Security are two of the most important. They allow you to seamlessly protect your users, upgrading their connections to HTTPS, and providing a guarantee that user data is never sent in the clear.
Why should you care? Consider this:
Delivering a web page over an unencrypted HTTP connection is more or less the same as handing an unsealed envelope to the first person you see on the street who looks like she's

Content injection is scary.

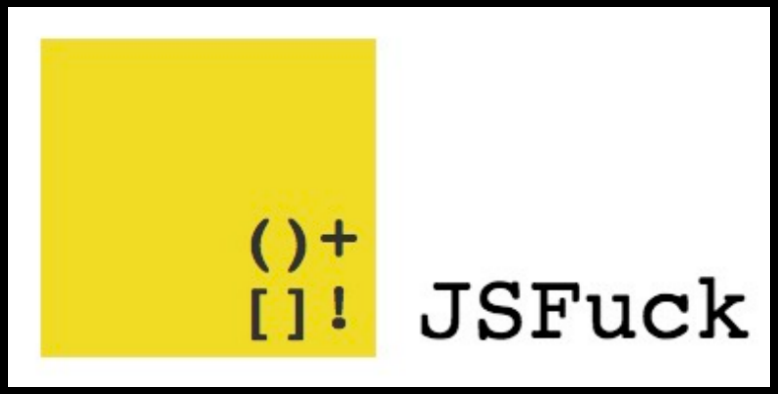
scheme://host:port

```
<script>
  beAwesome();
</script>
```

```
<script>
  beEvil();
</script>
```

```
                <script>
                  beAwesome();
                </script>

<!-- <p>Hello, {$name}!</p> -->
  <p>Hello, <script>
                  beEvil();
                </script>!</p>
```

```
<style>
 p { color: {{USER_COLOR}}; }
</style>
<p>
 Hello {{USER_NAME}}, view your
 <a href="{{USER_URL}}">Account</a>.
</p>
<script>
 var id = {{USER_ID}};
</script>
<!-- DEBUG: {{INFO}} -->
```


()+
[]!
JSFuck


```
[][(![]+[])[+[[+[]]]]+([][[]]+[])[+[[!+[]+!+[]+!+[]+!+[]+!+[]]]]+(![]+[])[+[[!+[]+!+[]]]]+(!![]+[])
[+[[+[]]]]+(!![]+[])[+[[!+[]+!+[]+!+[]]]]+(!![]+[])[+[[+!+[]]]]][([][(![]+[])[+[[+[]]]]+([][[]]+[])
[+[[!+[]+!+[]+!+[]+!+[]+!+[]]]]+(![]+[])[+[[!+[]+!+[]]]]+(!![]+[])[+[[+[]]]]+(!![]+[])[+[[!+[]+!+[]
+!+[]]]]+(!![]+[])[+[[+!+[]]]]+[])[+[[!+[]+!+[]+!+[]+!+[]]]]+(![]+[])[+[[!+[]+!+[]]]]+(!![]+[])
+!+[]+!+[]+!+[]+!+[]]]]+(![]+[])[+[[!+[]+!+[]]]]+(!![]+[])[+[[+[]]]]+(!![]+[])[+[[!+[]+!+[]+!+[]]]]
+(!![]+[])[+[[+!+[]]]]+[])[+[[!+[]+!+[]+!+[]+!+[]+!+[]+!+[]+!+[]]]]+([][[]]+[])[+[[+!+[]]]]+(![]+[])[+
[[!+[]+!+[]+!+[]]]]+(!![]+[])[+[[+[]]]]+(!![]+[])[+[[+!+[]]]]+([][[]]+[])[+[[+[]]]]+([][(![]+[])[+
[[+[]]]]+([][[]]+[])[+[[!+[]+!+[]+!+[]+!+[]+!+[]]]]+(![]+[])[+[[!+[]+!+[]]]]+(!![]+[])[+[[+[]]]]+
(!![]+[])[+[[!+[]+!+[]+!+[]]]]+(!![]+[])[+[[+!+[]]]]+[])[+[[!+[]+!+[]+!+[]+!+[]+!+[]+!+[]+!+[]]]]+
([][(![]+[])[+[[+[]]]]+([][[]]+[])[+[[!+[]+!+[]+!+[]+!+[]+!+[]+!+[]]]]+(![]+[])[+[[!+[]+!+[]]]]+(!![]+
[])[+[[+[]]]]+(!![]+[])[+[[!+[]+!+[]+!+[]]]]+(!![]+[])[+[[+!+[]]]]+[])[+[[!+[]+!+[]+!+[]+!+[]+!+[]+!+[]
+!+[]]]]+(!![]+[])[+[[+!+[]]]]((![]+[])[+[[+!+[]]]]+(![]+[])[+[[!+[]+!+[]]]]+(!![]+[])[+[[!+[]+!+
[]+!+[]]]]+(!![]+[])[+[[+!+[]]]]+(!![]+[])[+[[+[]]]]+([][(![]+[])[+[[+[]]]]+([][[]]+[])[+[[!+[]+!+
[]+!+[]+!+[]+!+[]]]]+(![]+[])[+[[!+[]+!+[]]]]+(!![]+[])[+[[+[]]]]+(!![]+[])[+[[!+[]+!+[]+!+[]]]]+
(!![]+[])[+[[+!+[]]]]+[])[+[[!+[]+!+[]]]+[[!+[]+!+[]+!+[]+!+[]+!+[]]]+[+!+[]]+([][(![]+[])[+[[+[]]]]+
([][[]]+[])[+[[!+[]+!+[]+!+[]+!+[]+!+[]]]]+(![]+[])[+[[!+[]+!+[]]]]+(!![]+[])[+[[+[]]]]+(!![]+[])[+
[[!+[]+!+[]+!+[]]]]+(!![]+[])[+[[+!+[]]]]+[])[+[[!+[]+!+[]+!+[]+!+[]+!+[]+!+[]+!+[]+!+[]]]])()
```

JSFuck

()+
[]!

```
[][(![]+[])[+[[+[]]]]+([][[]]+[])[+[[!+[]+!+[]+!+[]+!+[]+!+[]]]]+(![]+[])[+[[!+[]+!+[]]]]+(!![]+[])
[+[[+[]]]]+(!![]+[])[+[[!+[]+!+[]+!+[]]]]+(!![]+[])[+[[+!+[]]]]][([][(![]+[])[+[[+[]]]]+([][[]]+[])
[+[[!+[]+!+[]+!+[]+!+[]+!+[]]]]+(![]+[])[+[[!+[]+!+[]]]]+(!![]+[])[+[[+[]]]]+(!![]+[])[+[[!+[]+!+[]
+!+[]]]]+(!![]+[])[+[[+!+[]]]]]+[])[+[[!+[]+!+[]+!+[]]]]+([][(![]+[])[+[[+[]]]]+([][[]]+[])[+[[!+[]
+!+[]+!+[]+!+[]+!+[]]]]+(![]+[])[+[[!+[]+!+[]]]]+(!![]+[])[+[[+[]]]]+(!![]+[])[+[[!+[]+!+[]+!+[]]]]
+(!![]+[])[+[[+!+[]]]]]+[])[+[[!+[]+!+[]+!+[]+!+[]+!+[]+!+[]]]]+([][[]]+[])[+[[+!+[]]]]+(![]+[])[+
[[!+[]+!+[]+!+[]]]]+(!![]+[])[+[[+[]]]]+(!![]+[])[+[[+!+[]]]]+([][[]]+[])[+[[+[]]]]+([][(![]+[])[+
[[+[]]]]+([][[]]+[])[+[[!+[]+!+[]+!+[]+!+[]+!+[]]]]+(![]+[])[+[[!+[]+!+[]]]]+(!![]+[])[+[[+[]]]]+
(!![]+[])[+[[!+[]+!+[]+!+[]]]]+(!![]+[])[+[[+!+[]]]]]+[])[+[[!+[]+!+[]+!+[]]]]+(!![]+[])[+[[+[]]]]+
([][(![]+[])[+[[+[]]]]+([][[]]+[])[+[[!+[]+!+[]+!+[]+!+[]+!+[]]]]+(![]+[])[+[[!+[]+!+[]]]]+(!![]+
[])[+[[+[]]]]+(!![]+[])[+[[!+[]+!+[]+!+[]]]]+(!![]+[])[+[[+!+[]]]]]+[])[+[[!+[]+!+[]+!+[]+!+[]+!+[]
+!+[]]]]+(!![]+[])[+[[+!+[]]]]]((![]+[])[+[[+!+[]]]]+(![]+[])[+[[!+[]+!+[]]]]+(!![]+[])[+[[!+[]+!+
[]+!+[]]]]+(!![]+[])[+[[+!+[]]]]+(!![]+[])[+[[+[]]]]+([][(![]+[])[+[[+[]]]]+([][[]]+[])[+[[!+[]+!+
[]+!+[]+!+[]+!+[]]]]+(![]+[])[+[[!+[]+!+[]]]]+(!![]+[])[+[[+[]]]]+(!![]+[])[+[[!+[]+!+[]+!+[]]]]+
(!![]+[])[+[[+!+[]]]]]+[])[+[[+!+[]]]+[[!+[]+!+[]+!+[]+!+[]+!+[]]]]+[+!+[]]+([][(![]+[])[+[[+[]]]]+
([][[]]+[])[+[[!+[]+!+[]+!+[]+!+[]+!+[]]]]+(![]+[])[+[[!+[]+!+[]]]]+(!![]+[])[+[[+[]]]]+(!![]+[])[+
[[!+[]+!+[]+!+[]]]]+(!![]+[])[+[[+!+[]]]]]+[])[+[[+!+[]]]+[[!+[]+!+[]+!+[]+!+[]+!+[]+!+[]]]])()
```

alert(1);


XSS Filter Evasion Cheat Shee
https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet
Originally found by Begeek (but cleaned up and shortened to work in all browsers), this XSS vector uses the relaxed rendering engine to create our XSS vector within an IMG tag that should be encapsulated within quotes. I assume this was originally meant to correct sloppy coding. This would make it significantly more difficult to correctly parse apart an HTML tag:
<IMG """><SCRIPT>alert("XSS")</SCRIPT>">
fromCharCode
if no quotes of any kind are allowed you can eval() a fromCharCode in JavaScript to create any XSS vector you need:
<IMG SRC=javascript:alert(String.fromCharCode(88,83,83))>
https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet
goo.gl/XE0aW
all of the XSS examples that use a javascript: directive inside of an <IMG tag will not work in Firefox or Netscape 8.1+ in the Gecko rendering engine mode). Use the XSS Calculator for more information:
Long UTF-8 Unicode encoding without semicolons
This is often effective in XSS that attempts to look for "&#XX;", since most people don't know about padding - up to 7 numeric characters total. This is also useful against people who decode against strings like $tmp_string =~ s/.*\&#(\d+);.*/$1/; which incorrectly assumes a semicolon is required to terminate a html encoded string (I've seen this in the wild):
<IMG SRC=&#0000106&#0000097&#0000118&#0000097&#0000115&#0000099&#0000114&#0000105&#0000112&#0000116&#0000058&#0000097&
#0000108&#0000101&#0000114&#0000116&#0000040&#0000039&#0000088&#0000083&#0000083&#0000039&#0000041>
Hex encoding without semicolons
This is also a viable XSS attack against the above string $tmp_string =~ s/.*\&#(\d+);.*/$1/; which assumes that there is a numeric character following the pound symbol - which is not true

"I discount the probability of perfection."

-Alex Russell

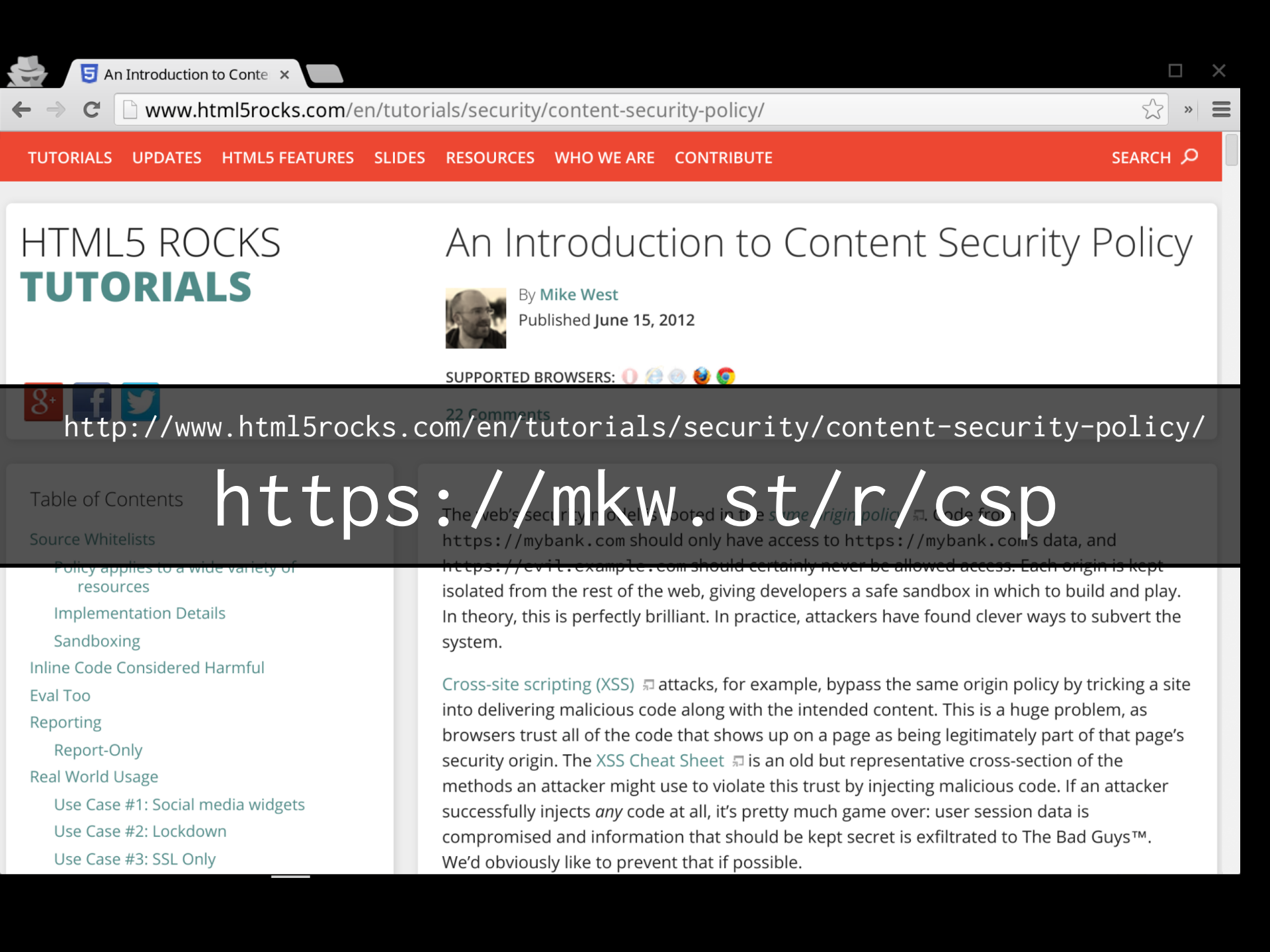
"We are all idiots with deadlines."

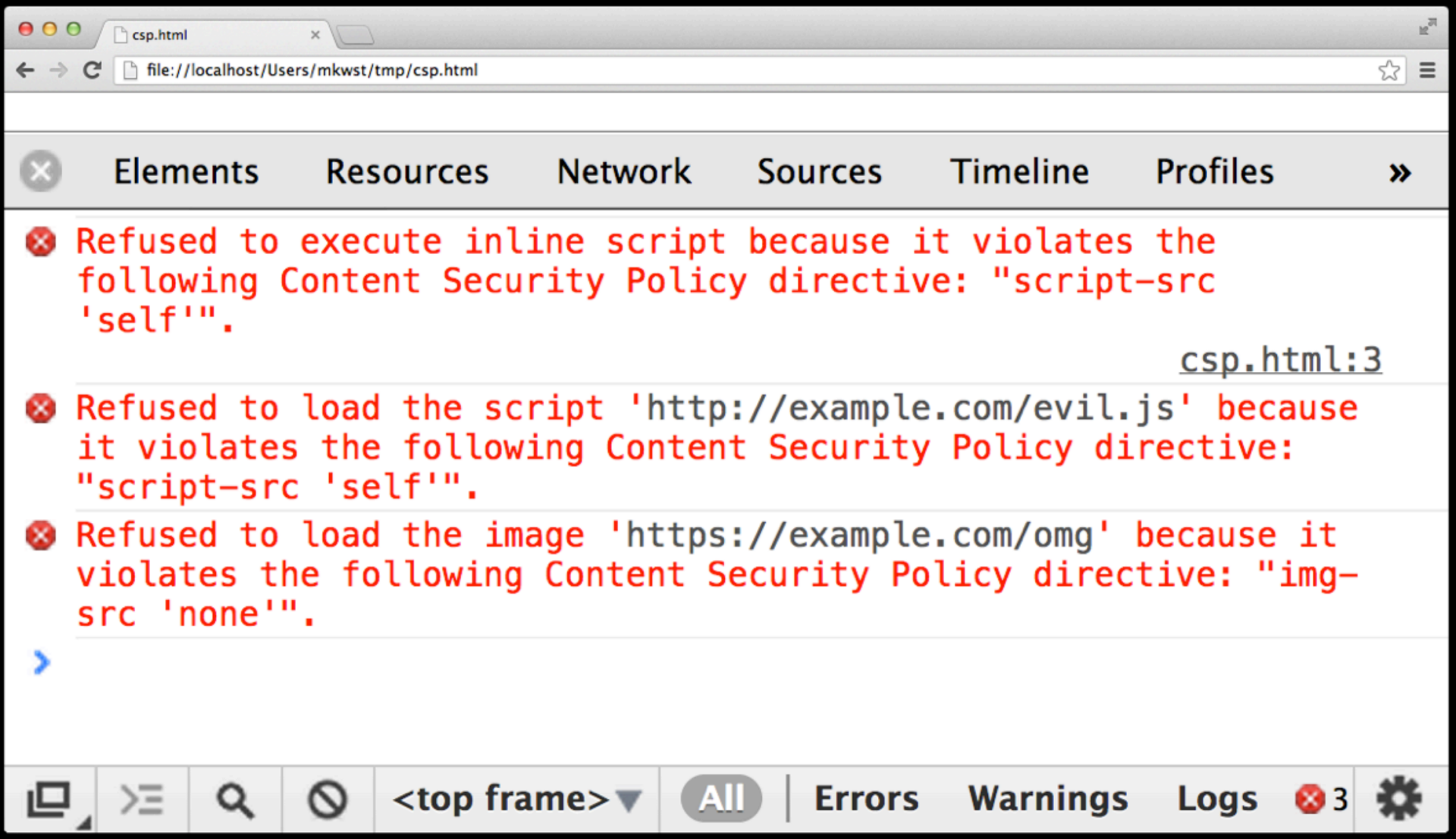
-Mike West

http://traumwerk.stanford.edu/philolog/2009/10/homers_odyssey_in_art_sirens_f.html

Principle of Least Privilege

http://w3.org/TR/CSP11


An Introduction to Conte
www.html5rocks.com/en/tutorials/security/content-security-policy/
TUTORIALS UPDATES HTML5 FEATURES SLIDES RESOURCES WHO WE ARE CONTRIBUTE SEARCH
HTML5 ROCKS
TUTORIALS
An Introduction to Content Security Policy
By Mike West
Published June 15, 2012
SUPPORTED BROWSERS:
22 Comments
http://www.html5rocks.com/en/tutorials/security/content-security-policy/
https://mkw.st/r/csp
Table of Contents
Source Whitelists
Policy applies to a wide variety of resources
Implementation Details
Sandboxing
Inline Code Considered Harmful
Eval Too
Reporting
Report-Only
Real World Usage
Use Case #1: Social media widgets
Use Case #2: Lockdown
Use Case #3: SSL Only
The web's security model is rooted in the same origin policy. Code from https://mybank.com should only have access to https://mybank.com's data, and https://evil.example.com should certainly never be allowed access. Each origin is kept isolated from the rest of the web, giving developers a safe sandbox in which to build and play. In theory, this is perfectly brilliant. In practice, attackers have found clever ways to subvert the system.
Cross-site scripting (XSS) attacks, for example, bypass the same origin policy by tricking a site into delivering malicious code along with the intended content. This is a huge problem, as browsers trust all of the code that shows up on a page as being legitimately part of that page's security origin. The XSS Cheat Sheet is an old but representative cross-section of the methods an attacker might use to violate this trust by injecting malicious code. If an attacker successfully injects any code at all, it's pretty much game over: user session data is compromised and information that should be kept secret is exfiltrated to The Bad Guys™. We'd obviously like to prevent that if possible.


csp.html
file://localhost/Users/mkwst/tmp/csp.html
Elements
Resources
Network
Sources
Timeline
Profiles
»
Refused to execute inline script because it violates the following Content Security Policy directive: "script-src 'self'".
csp.html:3
Refused to load the script 'http://example.com/evil.js' because it violates the following Content Security Policy directive: "script-src 'self'".
Refused to load the image 'https://example.com/omg' because it violates the following Content Security Policy directive: "img-src 'none'".
<top frame>
All
Errors
Warnings
Logs
3

```
Content-Security-Policy:
  default-src 'none';
  style-src   https://mikewestdotorg.hasacdn.net;
  frame-src   https://www.youtube.com
              https://www.speakerdeck.com;
  script-src  https://mikewestdotorg.hasacdn.net
              https://ssl.google-analytics.com;
  img-src     'self'
              https://mikewestdotorg.hasacdn.net
              https://ssl.google-analytics.com;
  font-src    https://mikewestdotorg.hasacdn.net
```

```
Content-Security-Policy:
  default-src ...;
  script-src ...;
  object-src ...;
  style-src ...;
  img-src ...;
  media-src ...;
  frame-src ...;
  font-src ...;
  connect-src ...;
  sandbox ...;
  report-uri https://example.com/reporter.cgi
```

```
Content-Security-Policy-Report-Only:
  default-src https:;
  report-uri https://example.com/csp-violations
```

```
{
  "csp-report": {
    "document-uri": "http://example.org/page.html",
    "referrer": "http://evil.example.com/haxor.html",
    "blocked-uri": "http://evil.example.com/img.png",
    "violated-directive": "default-src 'self'",
    "original-policy": "...",
    "source-file": "http://example.com/script.js",
    "line-number": 10,
    "column-number": 11,
  }
}
```

rick waldron
@rwaldron

Follow

CSP: designed by ignoring reality

Reply Retweet Favorite More

2	6
RETWEETS	FAVORITES

3:07 AM - 26 Aug 13 from New York, NY

https://twitter.com/rwaldron/status/371801007829041153

```
<script>
  function handleClick() { ... }
</script>
<button onclick="handleClick()">Click me!</button>
<a href="javascript:handleClick()">Click me!</a>
```

```
<!-- index.html -->
<script src="clickHandler.js"></script>
<button class="clckr">Click me!</button>
<a href="#" class="clckr">Click me!</a>

<!-- clickHandler.js -->
function handleClick() {
  ...
}

function init() {
  for (var e in document.querySelectorAll('.clckr'))
    e.addEventListener('click', handleClick);
}
```

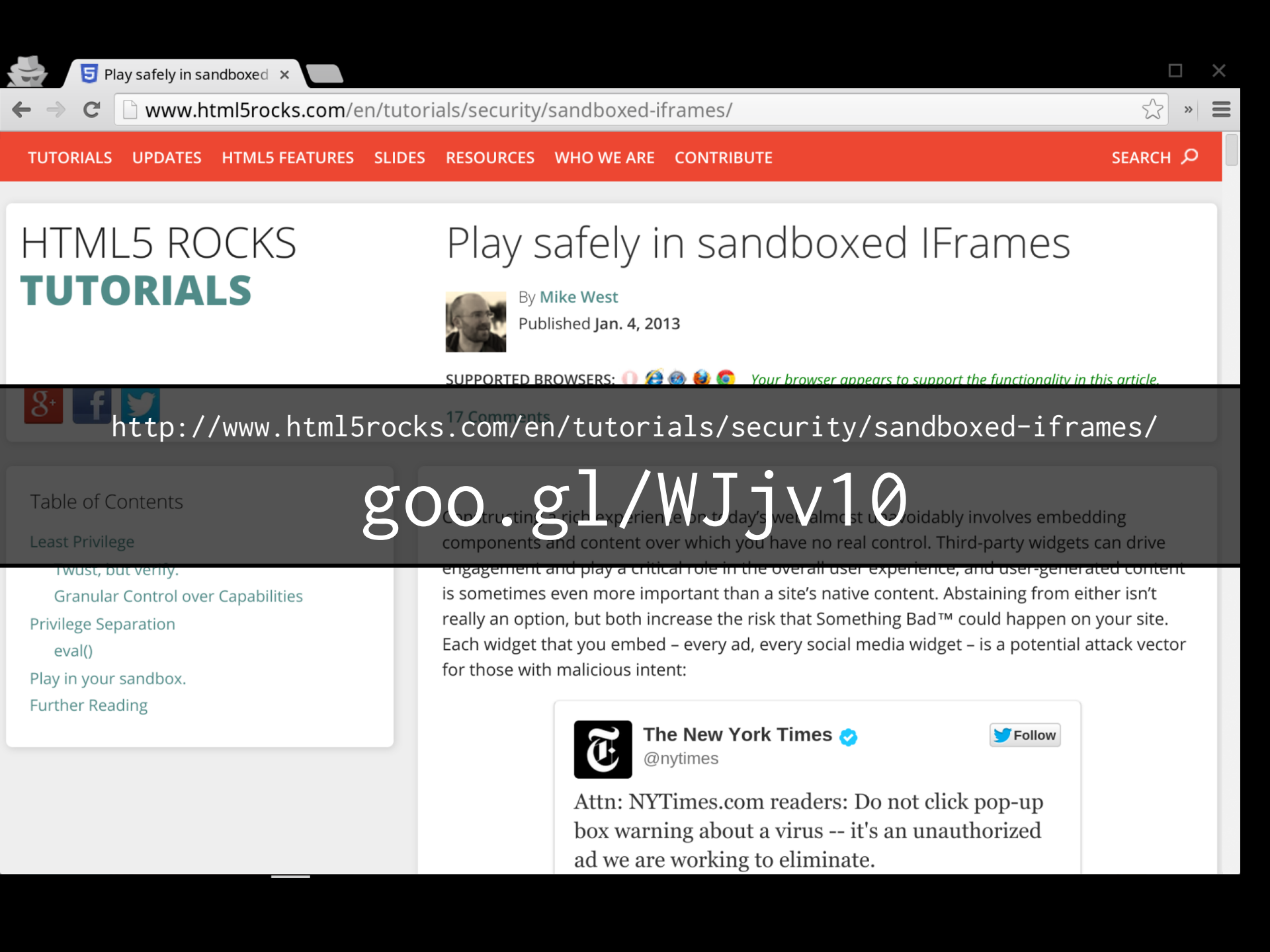
```
Content-Security-Policy:
  script-src 'nonce-afbvjn+afpo-j1qer';


<button class="clckr">Click me!</button>
<a href="#" class="clckr">Click me!</a>
<script nonce="oafbvjn+afpo-j1qer">
  function handleClick() { ... }
  function init() {
    var e;
    for (e in document.querySelectorAll('.clckr'))
      e.addEventListener('click', handleClick);
  }
</script>
```

`eval()` is evil?

```
<iframe src="page.html" sandbox></iframe>
<!--
  * Unique origin
  * No plugins.
  * No script.
  * No form submissions.
  * No top-level navigation.
  * No popups.
  * No autoplay.
  * No pointer lock.
  * No seamless iframes.
-->
```

```
<iframe src="page.html"
  sandbox="allow-forms allow-pointer-lock
           allow-popups allow-same-origin
           allow-scripts allow-top-navigation">
</iframe>
<!--
  * No plugins.
  * No seamless iframes.
-->
```

```
<!-- User-generated content? (in
     The Near Future™) -->
<iframe
 seamless
 srcdoc="<p>This is a comment!</p>"
 sandbox></iframe>
```


Play safely in sandboxed
www.html5rocks.com/en/tutorials/security/sandboxed-iframes/
TUTORIALS UPDATES HTML5 FEATURES SLIDES RESOURCES WHO WE ARE CONTRIBUTE SEARCH
HTML5 ROCKS
TUTORIALS
Play safely in sandboxed IFrames
By Mike West
Published Jan. 4, 2013
SUPPORTED BROWSERS:
Your browser appears to support the functionality in this article.
17 Comments
http://www.html5rocks.com/en/tutorials/security/sandboxed-iframes/
goo.gl/WJjv10
Table of Contents
Least Privilege
Trust, but verify.
Granular Control over Capabilities
Privilege Separation
eval()
Play in your sandbox.
Further Reading
engagement and play a critical role in the overall user experience, and user-generated content
is sometimes even more important than a site's native content. Abstaining from either isn't
really an option, but both increase the risk that Something Bad™ could happen on your site.
Each widget that you embed – every ad, every social media widget – is a potential attack vector
for those with malicious intent:
The New York Times
@nytimes
Follow
Attn: NYTimes.com readers: Do not click pop-up box warning about a virus -- it's an unauthorized ad we are working to eliminate.

https://mkw.st/r/goto13

Thanks!

Mike West
https://mikewest.org
G+: mkw.st/+
Twitter: @mikewest
Slides: https://mkw.st/r/goto13