# Sensitive Response

```
HTTP/1.1 200 OK
Content-Length: Number
Content-Type: application/json

{
  "phone" : "01*********",
  "token" : "**************"
}
```


**Mahmoud M. Awali**

@0xAwali

attacker

My Methodology

Try To Inject Origin Header e.g. Origin: http://me.com And If Response Contains Access-Control-Allow-Origin: http://me.com AND Access-Control-Allow-Credentials: True , There is CORS

- Writeup
- Writeup
- Writeup

```
GET /getInfo HTTP/1.1
Host: www.company.com
User-Agent: Mozilla/5.0
Referer: https://previous.com/path
Origin: http://me.com
```

attacker

My Methodology

**Try To Inject Origin Header With Your Domain e.g. http://company.com.me.com , http://Acompany.com , http://companyAcom , http://company.comA And null To Get CORS**

- Video
- Video
- Tweet

```
GET /getInfo HTTP/1.1
Host: www.company.com
User-Agent: Mozilla/5.0
Referer: https://previous.com/path
Origin: http://company.com.me.com
```

attacker

My Methodology

**Try To Inject Origin Header With Special Chars e.g. , & ' " ; ! $ ^ * ( ) + = ` ~ - _ = | { } %**
**OR Non Printable Chars e.g. %01-08 , %0b , %0c , %0e , %0f , %10-%1f AND %7f**

-  Blog
-  Writeup

```
GET /getInfo HTTP/1.1
Host: www.company.com
User-Agent: Mozilla/5.0
Referer: https://previous.com/path
Origin: http://me.com`.company.com
```

attacker

My Methodology

Try To Inject Origin Header e.g. Origin: http://me.com And If Response Contains Access-Control-Allow-Origin: * AND There Is Cache-Control: no-cache Header Try Use This POC

- Writeup
- Blog
- Tweet

```
<html>
<script>
var url = "https://www.company.com/getInfo";
fetch(url, {
   method: 'GET',
   cache: 'force-cache'
   });
</script>
</html>
```

attacker

My Methodology

Try To Use Web Cache Deception Attack , Add Static File e.g. nonexistent.css OR logo.png To Your Endpoint To Cache Sensitive Response

-  Slides
- Writeup
- Writeup
- Writeup
- Writeup

```
GET /getInfo/nonexistent.css HTTP/1.1
Host: www.company.com
User-Agent: Mozilla/5.0
Referer: https://previous.com/path
Origin: https://www.company.com
```

attacker

# My Methodology

**Try To Add jsonp OR callback To Any Sensitive Endpoints e.g. http://company.com/getInfo?jsonp=function To You Can Read The Response**

- Tweet
- Writeup

attacker

My Methodology

Try To Figure Out , There Is Cache-Control: no-cache Header In HTTP/1.1 Response OR Pragma: no-cache Header In HTTP/1.0 Response , If Yes There Is Issue Here

- Tweet

```
HTTP/1.1 200 OK
Content-Length: Number
Cache-Control: no-cache , no-store , must-revalidate
Content-Type: application/json

{
  "phone" : "01*********",
  "token" : "************"
}
```

attacker

My Methodology

## Try To Figure Out , There Is X-Frame-Option Header OR Not In The Response

-  Blog
-  Writeup
-  Writeup

```
HTTP/1.1 200 OK
Content-Length: Number
X-Frame-Option: SAMEORIGIN
Content-Type: application/json

{
  "phone" : "01*********",
  "token" : "************"
}
```

attacker

My Methodology

**There Are JSONP Endpoints OR JavaScript URL e.g. http://company.com/getInfo.js Contains Sensitive Data Based Only On Cookie , Try To Include It To Achieve XSSI**

-  Slides
-  Blog
-  Writeup

**Steps to produce :-**

**1 - Filter Burp Suite History To MIME type Script**
**2 - Lookout For JS Endpoints Without**
**X-Content-Type-Options: nosniff**
**3 - Search About**
**Sensitive Data**
**4 - Make Sure There Isn’t Authorization Headers**

attacker

My Methodology

Try To Figure Out , There Is postMessage API e.g. Window.postMessage("text", "*");
OR addEventListener("message", function(message){message.origin});

-  Blog
-  Blog
-  Blog
-  Blog
-  Blog

Steps to produce :-

1 - Browse To http://company.com/getInfo
2 - Right Clicking , Click On View Page Source
3 - Search About
    postMessage With *
    addEventListener With Argument-Function.origin

# Thank You

**Mahmoud M. Awali**

@0xAwali