

RSTCon CTF

Author: <Mihai Cujba / 0x435446> - <mihai.cujba@yahoo.com / 0x435446@gmail.com>

Summary

RSTCon CTF	1
Summary	1
<Motorul de cautare pentru vulnerabilitati> (<428>): <Web>	2
Proof of Flag	2
Summary	2
Proof of Solving	2
<Interviul> (<428>): <Misc>	5
Proof of Flag	5
Summary	5
Proof of Solving	5
<Mesajul de pe retea - I> (<50>): <Crypto>	6
Proof of Flag	6
RST{15ef7619e4db40f47be08e2e8c263ec2f688464b62598c713c1b68e0240fedce}	6
Summary	6
Proof of Solving	6
< Mesajul de pe retea - II> (<482>): <Crypto>	7
Proof of Flag	7
Summary	7
Proof of Solving	7

<Motorul de cautare pentru vulnerabilitati> (<428>): <Web>

Proof of Flag

RST{74343420409df9ada2608b0d8f070c1b7932f53a0c317cede3097e0c36a04352}

Summary


Folosirea unui token jwt vulnerabil din cauza secretului.

Proof of Solving

Prima data am incercat pe acel search-bar de pe prima pagina cateva tipuri de atac, pentru a verifica daca site-ul este vulnerabil. De aici am cautat sa verific daca este ceva in cookie-uri sau robots.txt, dar tot nimic, insa am gasit un token pus pe hidden in sursa paginii. L-am luat, l-am decodificat base64 ca sa pot vedea continutul si am vazut asta:


```
<input type="hidden" name="token" value="eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ1c2V5IiwiaWF0IjE2FuU2VhcmNoIjoibm8iLCJpYXQiOiJlY2MDYwNjU1NTYsImV4cCI6MTYwNjA2OTE1Nn0.D8IGxTJAV5LLkbcXpGKaaJNUajzYkmNil7sOWbRQJ2Y">
```

```
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ1c2V5IiwiaWF0IjE2FuU2VhcmNoIjoibm8iLCJpYXQiOiJlY2MDYwNjU1NTYsImV4cCI6MTYwNjA2OTE1Nn0.D8IGxTJAV5LLkbcXpGKaaJNUajzYkmNil7sOWbRQJ2Y
```

 For encoded binaries (like images, documents, etc.) use the file upload form a bit further down on this page.

UTF-8  Source character set.

☐ Decode each line separately (useful for multiple entries).

 Live mode OFF Decodes in real-time when you type or paste (supports only UTF-8 character set).

< DECODE > Decodes your data into the textarea below.

```
{"alg": "HS256", "sub": "user", "canSearch": "no", "iat": 1606065556, "exp": 1606069156}Lm16$Om
```

```
root@kali:~/Desktop/CTF/Competition/rst/vuln# python jwt2john.py eyJhbGciOiJIUzI1NiIsInR5cCI6IHYwZmVud2Vhcmljbm81LCJpYXQ0IjE2MDYwNjU1NTYsImV4cCI6MTYwNjA2OTE1Nm0uDBIGx7JA9SLlkbCkPgKaaJNUajzyKmN1  
OWBQR3ZY 2y/dee/nul1  
eyJhbGciOiJIUzI1NiIsInR5cCI6IHYwZmVud2Vhcmljbm81LCJpYXQ0IjE2MDYwNjU1NTYsImV4cCI6MTYwNjA2OTE1Nm0u0FC2o6C532405792cb91b7174629a6a33546a3cd892636297bb8e59b4502766  
root@kali:~/Desktop/CTF/Competition/rst/vuln#  
  
root@kali:~/Desktop/CTF/Competition/rst/vuln# john hash --wordlist:/usr/share/wordlists/rockyou.txt  
Using default input encoding: UTF-8  
Loaded 1 password hash (HMAC-SHA256 [password is key, SHA256 128/128 AVX 4x])  
No password hashes left to crack (see FAQ)  
root@kali:~/Desktop/CTF/Competition/rst/vuln# john --show hash  
?:livestrong  
  
1 password hash cracked, 0 left  
root@kali:~/Desktop/CTF/Competition/rst/vuln#
```

The diagram illustrates the structure and verification of a JWT token. It is divided into three main sections: **HEADER**, **PAYLOAD**, and **VERIFY SIGNATURE**.

- HEADER:** Contains the algorithm and token type. The example shows: `{ "alg": "HS256" }`.
- PAYLOAD:** Contains the data. The example shows: `{ "sub": "user", "canSearch": "yes", "iat": 1606065556, "exp": 1606069156 }`.
- VERIFY SIGNATURE:** Shows the process of verifying the signature. It uses the HMACSHA256 algorithm with the base64 encoded header and payload, and a secret key (labeled "Weak secret!"). The example shows: `HMACSHA256(base64UrlEncode(header) + "." + base64UrlEncode(payload), livestrong)`.

The final output is a JWT token: `eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJ1c2VyIiwiaWF0IjoxNjA2MDY1NTU2LCJleHAiOiE2MDYwNjkxNTZ9.HBSn7AEKJqT3cbyc3bUuT3M2lrF0012CdbIUocNi40o`.

Dupa ce l-am craftat am interceptat un request catre site in care am pus “flag” in search-bar, am inlocuit token-ul cu cel generat si am scos flag-ul.

Burp Suite Community Edition v2020.8.1 - Temporary Project

Burp Project Intruder Repeater Window Help

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options

1

Send Cancel < >

Target: http://vps-b40e6212.vps.ovh.net

Request

Raw Params Headers Hex

```

1 POST /searchengine/ HTTP/1.1
2 Host: vps-b40e6212.vps.ovh.net
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:83.0)
  Gecko/20100101 Firefox/83.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,image/web
  p,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 171
9 Origin: http://vps-b40e6212.vps.ovh.net
10 Connection: close
11 Referer: http://vps-b40e6212.vps.ovh.net/searchengine/
12 Cookie: ln=guest
13 Upgrade-Insecure-Requests: 1
14
15 search=flag&token=
  eyJhbGciOiJIUzI1NiIsInR5cGU6IiwiOiJlc2VybWVhcnNoIjoieWVzIi
  viaWF0IjoibWVhcnNoIj0iLCJleNA0yE2MDTmWjksNTI9.HB9n7AEKJqT3cbyc
  3bUuT3N21rF0012CdbIUscN40a

```

Response

Raw Headers Hex Render

```

1 HTTP/1.1 200 OK
2 Date: Sun, 22 Nov 2020 17:28:46 GMT
3 Server: Apache/2.4.37 (centos)
4 X-Powered-By: PHP/7.2.24
5 Connection: close
6 Content-Type: text/html; charset=UTF-8
7 Content-Length: 1604
8
9 <html>
10 <head>
11 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
12 <meta name="viewport" content="width=device-width, initial-scale=1" />
13 <meta http-equiv="X-UA-Compatible" content="IE=edge" />
14 <meta name="author" content="colorlib.com">
15 <link href="https://fonts.googleapis.com/css?family=Poppins:400,500,700" rel="stylesheet" />
16 <link href="css/main.css" rel="stylesheet" />
17 <title>
  Motorul de cautare pentru vulnerabilitati
18 </title>
19 </head>
20 <body>
21 <div class="s013">
22 <form method="post">
23 <fieldset>
24 <legend>
  GASESTE RAPID VULNERABILITATEA TA<br />
25 <small>
  ESTI(74343420405df9ada260b0d8f070c1b7932f53a0c317cde3097e0c16a04350)
26 </small>
27 </legend>
28 </fieldset>
29 <div class="inner-form">
30 <div class="left">
31 <div class="input-wrap first">
32 <div class="input-field first">
33 <label>
  CE
34 </label>
35 <input type="text" placeholder="ex: xss, sql, rce, flag, etc." name="search" />
36 </div>
37 </div>
38 <input type="hidden" name="token" value="eyJhbGciOiJIUzI1NiIsInR5cGU6IiwiOiJlc2VybWVhcnNoIjoieWVzIi
  viaWF0IjoibWVhcnNoIj0iLCJleNA0yE2MDTmWjksNTI9.HB9n7AEKJqT3cbyc
  3bUuT3N21rF0012CdbIUscN40a" />
39 <button class="btn-search" type="submit">
  CAUTA
40 </button>

```

<Interviul> (<428>): <Misc>

Proof of Flag

RST{e1a8697df258e73f9b430acd3c0d74dcf8b9c4025ab4aaa8abdab2b8c19f9718}

Summary

DNS Records

Proof of Solving

Cand am citit prima data descrierea challenge-ului, am crezut ca trebuie sa caut recursiv prin toate paginile de pe rstcon.com si flag-ul se va gasi pe undeva printr-un comentariu, dar aparent nu era asa. Dupa ce m-am gandit putin am incercat sa verific DNS Records. Am intrat pe <https://www.digwebinterface.com>, am pus site-ul in textbox, am dat pe ANY Records si am gasit flag-ul in cele TXT.

Hostnames or IP addresses:
rstcon.com

Type:
ANY

Nameservers:
☒ Resolver: Default
☐ All
☐ Authoritative
☐ NIC
☐ Specify myself:

Options:
☐ Show command
☐ Colorize output
☐ Stats
☐ Trace
☐ Sort alphabetically
☐ Short
☐ No recursive
☐ Only first nameserver
☐ Compare output
☐ Save to file
☐ Show IP geolocation
☐ DNSSEC

Dig Fix Reset form

rstcon.com@8.8.4.4 (Default):

rstcon.com.	299	IN	A	192.0.78.24
rstcon.com.	299	IN	A	192.0.78.25
rstcon.com.	21599	IN	NS	ns1.wordpress.com.
rstcon.com.	21599	IN	NS	ns2.wordpress.com.
rstcon.com.	21599	IN	NS	ns3.wordpress.com.
rstcon.com.	21599	IN	SOA	ns1.wordpress.com. hostmaster.wordpress.com. 2005071858 14400 7200 604800 300
rstcon.com.	3599	IN	MX	0 smtp-fwd.wordpress.com.
rstcon.com.	3599	IN	TXT	"RST{e1a8697df258e73f9b430acd3c0d74dcf8b9c4025ab4aaa8abdab2b8c19f9718}"

<Mesajul de pe retea - l> (<50>): <Crypto>

Proof of Flag

RST{15ef7619e4db40f47be08e2e8c263ec2f688464b62598c713c1b68e0240fedce}

Summary

Am primit un mesaj codificat.

Proof of Solving

De cand am vazut padding-ul de la base64 am incercat sa vad daca as putea decodifica mesajul cu aceasta metoda. Dupa decodificare am primit in mesaj ce parea scris de la dreapta la stanga, asa ca am facut un rev() pe el si a iesit cam asa:

Uvorxrgzir! Uozt-fo vhg

IHG{15vu7619v4wy40u47yv08v2v8x263vx2u688464y62598x713x1y68v0240uvwxv}.

Recipe

From Base64

Alphabet
A-Za-z0-9+/=

☒ Remove non-alphabet chars

Reverse

By
Character

Input

Ln12eHd2dTAA0Hj820DZ5MXgzHTd40Ok1MjZ5NDY00g2dT34djM2Mng4djJ200B2eTc0dTAA0eXc0djKxNjd1djUxe0dIS5B2Z2h2IG9mLXR6b1UgIXJpemdyeHJvd1U=

Output

Uvorxrgzir! Uozt-fo vhg IHG{15vu7619v4wy40u47yv08v2v8x263vx2u688464y62598x713x1y68v0240uvwxv}.

De aici am incercat diferite metode de criptare pe care se poate face usor bruteforce si am gasit ca este criptat folosind affine.

Search for a tool

★ SEARCH A TOOL ON DCODE BY KEYWORDS:

e.g. type boolean GO

Results

↑↓

↑↓

A=25, B=25

Felicitari! Flag-ul este
RST{15ef7619e4db40f47be08e2e8c263ec2f688464b62598c713c1b68e0240fedce}

AFFINE DECODER

★ AFFINE CIPHERTEXT

Uvorxrgzir! Uozt-fo vhg
IHG{15vu7619v4wy40u47yv08v2v8x263vx2u688464y62598x713x1y68v0240uvwxv}

★ EXPECTED PLAINTEXT LANGUAGE English

★ ALPHABET ABCDEFGHIJKLMNOPQRSTUVWXYZ

AUTOMATIC BRUTE FORCE DECRYPTION

< Mesajul de pe retea - II> (<482>): <Crypto>

Proof of Flag

RST{84aa7743e2633f55a8a9d9322cd30d664e8440a4341f08b00f72ceb5e411133a}

Summary

Asemanator cu Mesajul de pe retea – I, am primit un mesaj encodat.

Proof of Solving

Pentru inceput am incercat sa trec mesajul prin base32, base58, base62 si base64, dar fara succes. De aici m-am gandit sa fac acelasi lucru, dar facut un rev() peste textul initial; tot nimic.

Am mai stat putin sa ma gandesc si am incercat caesar peste text, iar la ROT 13 am reusit sa scot un text in clar, folosind base32 peste.

Textul avea un padding la inceput, deci am dedus ca este scris de la dreapta la stanga. Am facut un rev() pe el si arata a text encodat cu base64. L-am decodificat si mi-a dat de brainfuck (literalmente 😊).

The image shows a web application for performing ROT13 encryption and decoding. The interface is divided into several sections:

- Recipe:** A sidebar on the left containing settings for the cipher. It includes a 'ROT13' button, checkboxes for 'Rotate lower case chars' and 'Rotate upper case chars' (both checked), an 'Amount' dropdown set to '13', a 'From Base64' section with an 'Alphabet' dropdown set to 'A-Za-z0-9+/=', and a 'Reverse' section with a 'By Character' dropdown.
- Input:** A large text area on the right containing a long string of base64-encoded text. The status bar indicates 'length: 960' and 'lines: 1'.
- Output:** A text area at the bottom showing the result of the operation. The status bar indicates 'time: 2ms', 'length: 720', and 'lines: 1'. The output is a brainfuck program starting with '==gLB0FPrsyK+0SLb1SL+4SL+0FPrs1PtsVLT4iLrsiLu4SLt0iLr4TX8siPt0yW4SL+0FPrs1PtsVLT4yKrsyK+0FPr4TLts1Lt0SLusyKu0iPdxzKr4T'.

[illegible][illegible]