Fundamental Skills - Windows Command Line Usage

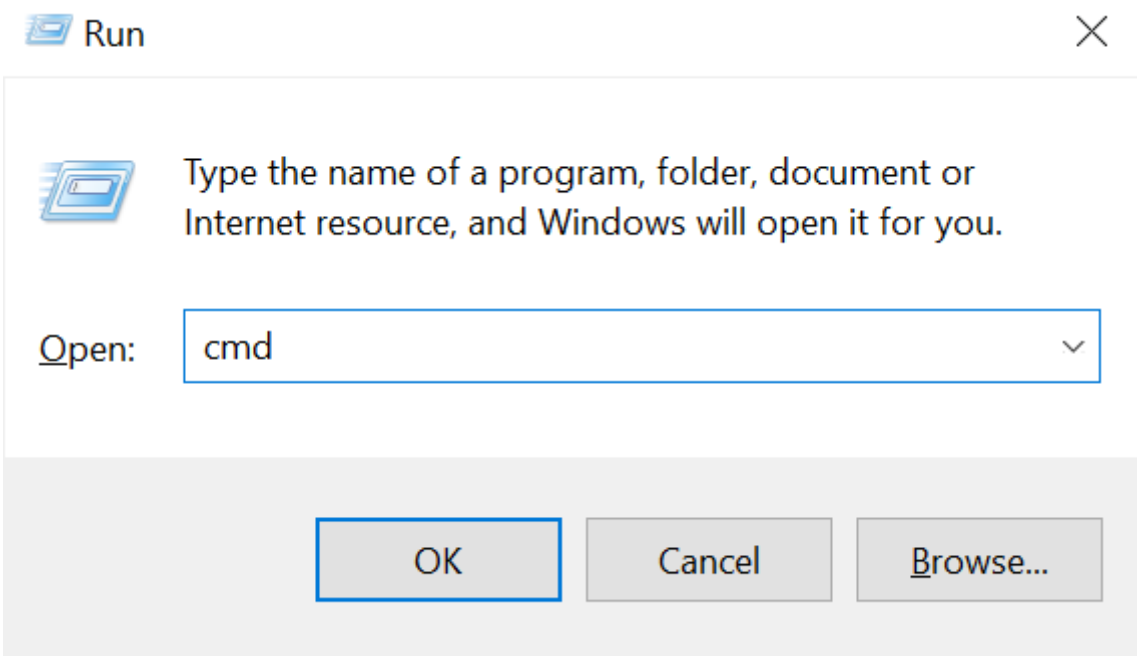
Category	Experience Level	Author
Windows	Complete Beginner	Mac Goodwin

Contents

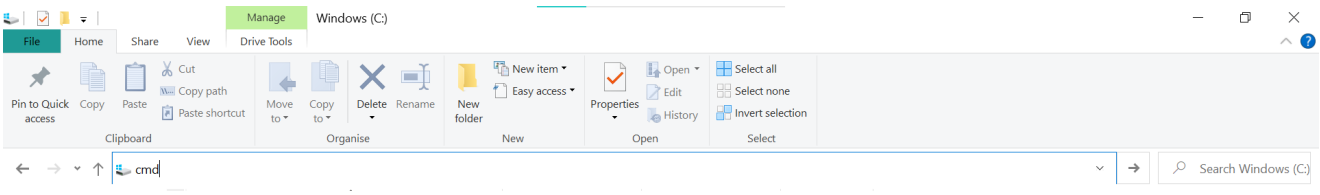
Intro

While windows is a primarily Graphical User Interface (GUI) based Operating System, it does have some command line interfaces (CLIs) built in. Command Prompt (or CMD) is the most basic Windows Shell. PowerShell is, as the name suggests, a much more powerful piece of software that includes not only a CLI but also a scripting language.

You can launch CMD by typing 'Command Prompt' into the start menu, or with the Run menu (`Windows Key + R`):



By default, it will open in your user's directory (usually `C:\Users\username`). To open it in a specific folder, navigate to the folder and type `cmd` into the location bar:



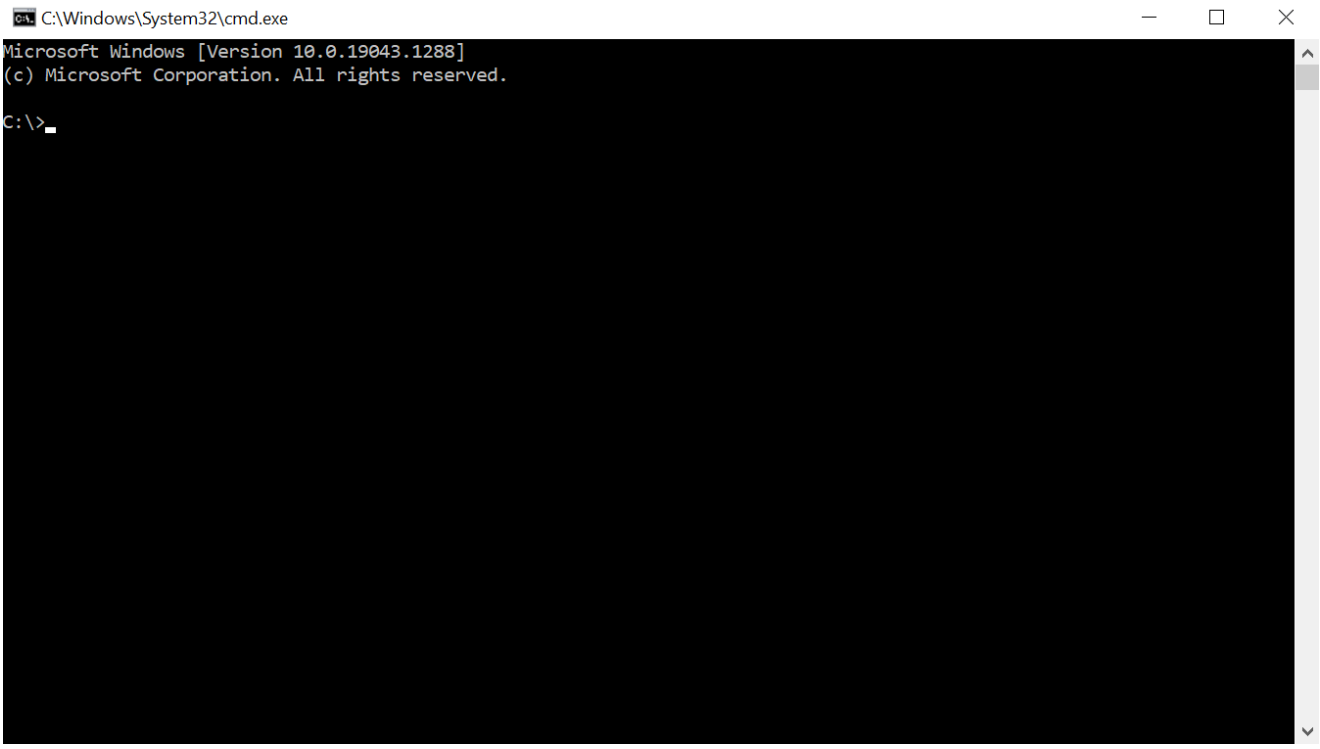
You can do the same to launch PowerShell - just type `powershell` instead.

If you're trying to launch a new process, perhaps through a CLI, you may sometimes need to specify the full path to the executable. By default:

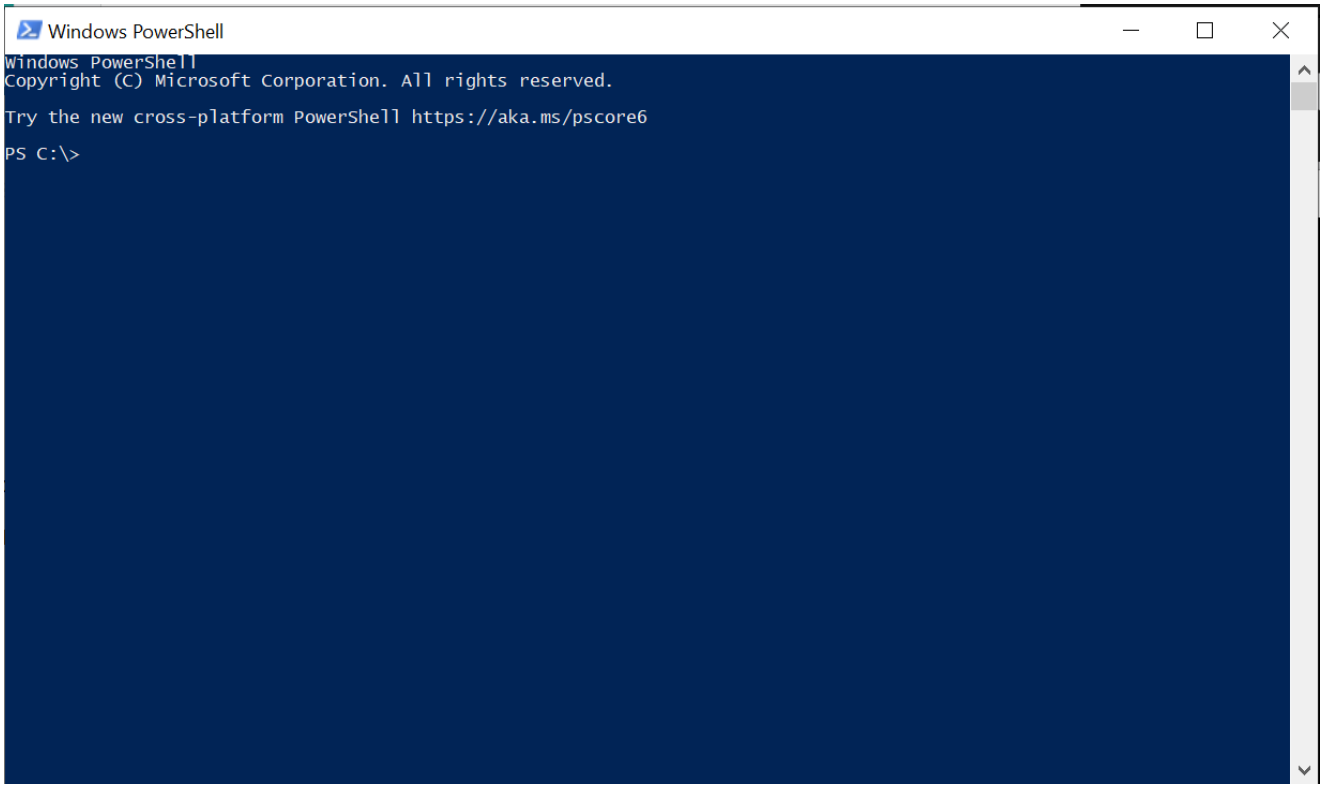
- Command Prompt is located at `C:\Windows\System32\cmd.exe`
- PowerShell is located at `C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe`

Which Shell Am I In?

If you're launching CMD/PowerShell from a GUI, it's easy to tell by the appearance of the shell which you're using. Command Prompt looks like this:



And PowerShell looks like this:

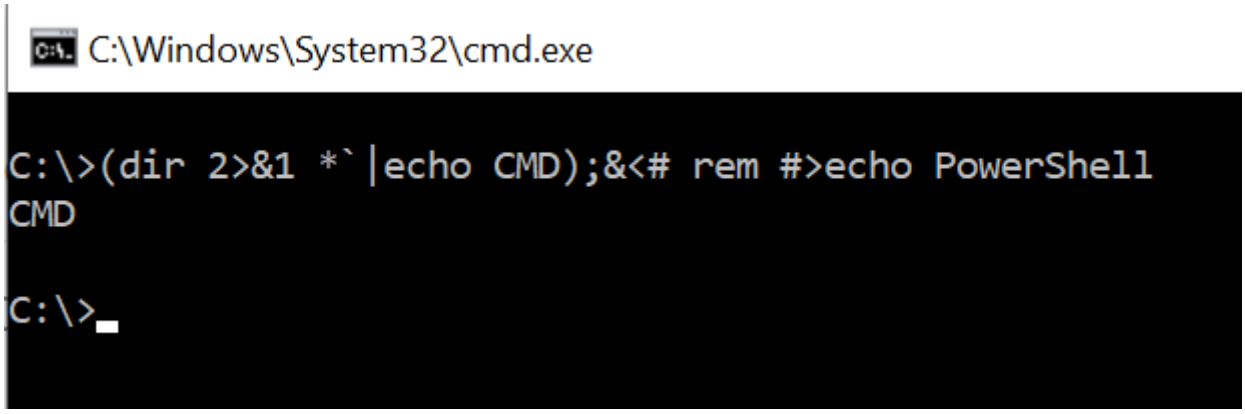


If you've returned a shell remotely, it can be a little harder to tell. As you can see from the image above, PowerShell shells often have the prefix `PS` before the drive location.

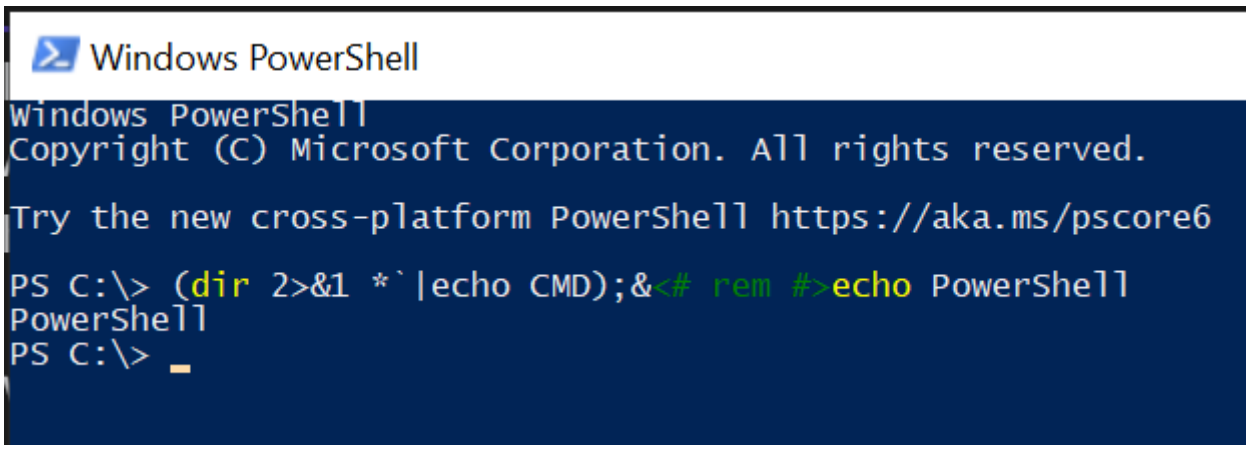
There is a definite way to tell by running a command - type the following text in your terminal prompt:

```
(dir 2>&1 *`|echo CMD);&<# rem #>echo PowerShell
```

The shell will tell you what it is. For example, in CMD:



And in powershell:



Source: https://stackoverflow.com/questions/34471956/how-to-determine-if-im-in-powershell-or-cmd

Command Prompt

Here are some of the most common things you'll do in Command Prompt.

To switch drive (e.g. from `C:` to a removable drive `E:`), type the letter of the drive followed by a colon:

```
C:>E:

E:>
```

To change directory, type `cd` (as in Unix):

```
C:>cd c:\users\username
```

As with Unix, you can change directory to one that's relative to you (e.g. with `..\directory` to reach a directory one level up) or specify a full path. Unlike Unix systems, the Windows file system is not case-sensitive - this makes it easier to type filepaths, and you can use `Tab` to autocomplete as usual.

To list files in a directory, use `dir`:

```
C:\Users\mac>dir
 Volume in drive C is Windows
 Volume Serial Number is 6ADF-C330

 Directory of C:\Users\mac

12/10/2021  13:16    <DIR>          .
```

```
12/10/2021  13:16    <DIR>          ..
20/10/2021  23:59             4,929 .bash_history
12/10/2021  13:16    <DIR>          .conda
05/03/2020  10:42    <DIR>          .config
05/02/2019  10:10    <DIR>          .eclipse
14/11/2018  11:34               285 .gitconfig
29/11/2019  14:02    <DIR>          .IntelliJIdea2019.3
06/10/2021  10:44    <DIR>          .ipython
01/10/2021  11:04    <DIR>          .isabelle
10/03/2019  19:01    <DIR>          .m2
06/10/2021  10:44    <DIR>          .matplotlib
29/11/2019  14:22    <DIR>          .p2
11/12/2019  14:07    <DIR>          .PyCharm2019.3
12/10/2021  13:29    <DIR>          .pylint.d
24/02/2019  20:36    <DIR>          .RubyMine2018.3
20/10/2021  11:58    <DIR>          .spyder-py3
01/10/2021  11:32    <DIR>          .ssh
29/11/2018  13:32    <DIR>          .tooling
19/08/2021  20:53    <DIR>          .VirtualBox

...[etc]...
```

To view the contents of an ASCII file, use `type`:

```
C:\Users\mac>dir
12/10/2021  13:16    <DIR>          .
12/10/2021  13:16    <DIR>          ..
...
14/11/2018  11:34                88 test.txt

C:\Users\mac>type test.txt
Hello world!
```

To find the location of a file, use `where`:

```
C:>where cmd.exe
C:\Windows\System32\cmd.exe
```

To run an executable file, just type the full path to the file:

```
C:>C:\windows\System32\WindowsPowerShell\v1.0\powershell.exe
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:>
```

To copy text, highlight it and press `Ctrl + C`. To paste into the terminal window, right-click.

To clear the terminal, type `cls`.

To see which user you are, type `whoami`:

```
C:>whoami
desktop-a18hl5m\mac
```

To see extra information, type `whoami /all`.

To see running processes, use `tasklist`:

```
C:\Users\mac>tasklist

Image Name                     PID Session Name        Session#    Mem Usage
========================= ======== ================ =========== ============
System Idle Process              0 Services                   0          8 K
System                           4 Services                   0      1,552 K
Registry                       124 Services                   0     30,440 K
smss.exe                       520 Services                   0        144 K
csrss.exe                      792 Services                   0      2,684 K
wininit.exe                    884 Services                   0      1,588 K

...
```

To run a command as another user, use `runas`:

```
C:>runas /user:USERNAME [PATH_TO_EXE]
```

For example, to run command prompt as the `administrator` user: `runas /user:administrator "C:\windows\system32\cmd.exe"` - this command is useful in

situations such as privilege escalation, but will require the user's password.

Powershell

PowerShell some parallels with bash commands - under the hood, PowerShell uses *cmdlets* (basically functions) like `Get-Content` to perform tasks, but often *aliases* these functions to easier-to-type commands. Therefore, some of PowerShell's most basic functionality may feel familiar if you're used to Bash:

- `ls` to list the contents of the current directory (actually an alias of `Get-ChildItem`)
- `cat` to read the contents of a file (actually an alias of `Get-Content`)
- `cp` to copy a file (actually an alias of `Copy-Item`)
- `curl` to make a HTTP request (actually an alias of `Invoke-WebRequest`)

PowerShell supports scripting - a series of commands can be saved to a `.ps1` file, and run by typing the path to the file:

```
PS C:\Users\mac\Documents> cat .\whoami.ps1
whoami
Get-Content .\script.ps1
PS C:\Users\mac\Documents> .\whoami.ps1
desktop-a18hl5m\mac
whoami
Get-Content .\script.ps1
```

Powershell can run an exe, like CMD can, using the `&` operator:

```
PS C:\Users\mac\Documents> & 'C:\Windows\System32\cmd.exe'
Microsoft Windows [Version 10.0.19043.1288]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mac\Documents>(dir 2>&1 *`|echo CMD);&<# rem #>echo PowerShell
CMD
```

PowerShell is also capable of loading .NET modules and external scripts with the `IEX` command, but we won't touch on this here.

Going Further

This article has a nice list of Command Prompt commands, by category. Many of these may be useful for privilege escalation and post-exploitation enumeration.

This article has a good table of basic PowerShell commands and aliases, and this article shows some more advanced commands used for system administration.