#Command Prompt and PowerShell

##What are command prompt and PowerShell

Command prompt was first introduced in Windows to automate routine tasks, like user account management or nightly backups.[^1] What does this mean for the average user? It means that people want access to command prompt because it has access to some of the most critical parts of your computer’s infrastructure. In most instances, programs run through command prompt won’t do much damage unless the goal of achieving administrative privileges has been achieved when a user has admin, also known as the root, privileges. They can do a lot of damage.

To illustrate some of the damage that a command prompt can do, you can check for all incoming and outgoing connections to your computer using simple commands. If a computer wants your information, you can use a PowerShell script to run an ipconfig and store the results in a document, then have that document sent to a specific IP address. If you’re the attacker, it’s more than likely your IP address.

##Why is using Command Prompt necessary?

Most programs that use some form of automation can do so easily by implementing simple PowerShell scripts. Companies use PowerShell for administrative purposes, and with the advent of cloud services, most commands that deal with administrative duties are executed through PowerShell commands. For instance, you can download cmdlets, applications to be used in tandem with PowerShell, to monitor and gain access to information concerning every single Outlook file attached to your organization’s workspace.

Several commands allow you to access critical information about your computer’s mechanical operations. Such commands include:

* HKEY\_CLASSES\_ROOT (HKCR) defines which applications are executed when a file is opened and contains information such as user interface information.[^2]
* HKEY\_CURRENT\_USER defines the currently logged-on user, including the folder, display, and control panel settings.[^3] This could be devastating if a bad actor wanted to steal a file from you and needed to know the path to set up a PowerShell script to open, copy, and send the file.

These are just a few of the commands at the CMD application's disposal and the PowerShell application, with more being found at [Microsoft](https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/windows-commands).

##What are root files

Root files are the core of any computer, and they contain files that pertain to system instructions and give definitions to the applications that can run. Why does it asks you where you want to install a program? Generally, it’s a path that the program executes from. CMD and PowerShell allow you to alter that path, see where files are located, and what files run. The programs will run in something akin to a file path “C:/WINDOWS for most Windows computers.” Don’t believe me? Try these steps:

* Look at the start or the use of search
* Type CMD or cmd
* In the space that opens in the new window, type echo %SYSTEMROOT%

Assuming your computer is set up like every default computer, it should display something akin to C:/WINDOWS or C:/Windows. In that location, files such as System and System 32 exist and run all the programs tied to the computer that run system programs and interact with everything connected to your computer.

For Linux, you can essentially do the same thing, but the hierarchy for Linux computers is more robust and separated, making it harder to intrude on Linux-based systems. The tree structure for a Linux root file system looks like this:

* /etc - This directory contains the files generally used in system administration and can include host and password information.[^4]
* /bin - This one is co-linked with /sbin and contains the files needed to mount and boot the machine. Editing these operations can get someone access to your hard drive or give them the ability to mount their own![^5]
* /dev - This cheeky devil contains the particular files for local devices, including but not limited to tape drives, printers, disk partitions, and terminals![^6]

##Do all malicious codes act through this?

No, but they can! More viruses come through in the form of rootkits, designed to have administrative control over your computer.

###What is a rootkit?

There are multiple types of rootkits[^7]:

* Hardware or firmware rootkits are designed to affect your hard drive, router, or system’s BIOS. This rootkit is what keyloggers use to monitor online activity and log your keystrokes.
* Bootloader rootkits are responsible for replacing what boots your computer with a malicious file and can lock you out of your computer.
* Memory rootkits are hidden in your RAM and use your computer to carry out malicious code until the next reset.

A famous example of a rootkit is the Stuxnet rootkit that struck in 2010 and was the first-ever perceived cyberweapon ever created.[^8]

#Works Cited

[^1]Gerend, J. (2022, January 4). *Windows commands*. Windows commands | Microsoft Docs. Retrieved April 14, 2022, from <https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/windows-commands>

[^2][^3]Cook, K. D., & Shashidhar, N. (2018). Using PowerShell to Capture and Compare Windows Registry and Live Memory Artifacts with Online Databases to Identify Suspect Files. *International Journal of Information Security Science*, *7*(2), 78–89.

[^4][^5][^6]IBM. (2021, December 3). *Root File System*. Root File System. Retrieved April 14, 2022, from <https://www.ibm.com/docs/pl/aix/7.1?topic=tree-root-file-system>

[^7][^8]Kaspersky. (2022, March 9). *What is rootkit – definition and explanation*. www.kaspersky.com. Retrieved April 14, 2022, from https://www.kaspersky.com/resource-center/definitions/what-is-rootkit