# WINZIP Case – Race condition

I’ve created 2 videos that demonstrate the Winzip race condition case:

* **Winzip\_part\_1\_locate\_exe.mp4** -> Locates the vulnerable exe file that is being written and executed from vulnerable location: C:\Windows\Temp\
* **Winzip\_part\_2\_race\_condition.mp4** -> Uses the forcecopy command to perform the race condition attack on the installation. And finally abuses the installation to run test.exe as SYSTEM.

# Winzip\_part\_1\_locate\_exe.mp4

This is the description for the first video: **Winzip\_part\_1\_locate\_exe.mp4.**

The aim is locate exe’s and dlls that we can abuse.

* First run process monitor 64 bit as administrator.
* Then deselect all the options except the file operations (folder icon) on top
* Select the filter to monitor the directory of interest (path begins with): C:\Windows\Temp
* Run command prompt (cmd.exe) as Administrator
* From the Administrator cmd escalate to SYSTEM using **psexec64 /s /i cmd** , from the C:\BIN directory
* Clear the process monitor. So its output not polluted from non-related programs.
* Run the winzip installation msi from the SYSTEM cmd. You can run the command: msiexec.exe /i winzip250-64.msi /q , to best simulate the SCCM installation.
* You may choose to opt out from running the executable to not pollute your process monitor output.
* Now save the filtered output of process monitor as PML file.
* Open the saved PML file and save it as csv file into C:\TEST
* Open the CSV file in Excel and save it as Excel workbook (xlsx).
* In Excel split the first column into multiple columns using Date-> Text To Columns. Use comma as a delimiter.
* Highlight the first row and make it Filter (Data->Filter).
* Resize the columns for better viewing.
* Click on the Path column and now you can inspect for exe and dll files that can be abused. Feel free to use other filters as well based on your need to better understand the situation.
* Locate the exe that we’re going to abuse: C:\Windows\Temp\CloseFAH.exe

Notice that there may be other subdirectories with exe’s and dlls created under C:\Windows\Temp\ , those could be writable by ordinary user but can be for example randomly generated. Aim for the low hanging fruit first for best exploitability. So exe’s and dlls right in C:\Windows\Temp. If nothing is found then tackle the subdirectories in C:\Windows\Temp for possible exploitation.

# Winzip\_part\_2\_race\_condition.mp4

This is the description for the first video: **Winzip\_part\_2\_race\_condition.mp4**

From the first part we have located a potentially vulnerable exe we want to attack. C:\Windows\Temp\CloseFAH.exe . Now we want to win the race with the winzip installation to substitute CloseFAH.exe with our test64.exe and make the installation run it as SYSTEM.

These are the steps:

* Run cmd.exe as normal (non-admin) user.
* Use forcecopy to repeat overwriting of C:\Windows\Temp\CloseFAH.exe with our test64.exe. This is the command: **forcecopy test64.exe C:\Windows\Temp\CloseFAH.exe**
* Run cmd.exe as administrator.
* From the administrator cmd.exe run **psexec64 /s /i cmd** , to launch cmd as SYSTEM. (You can close the administrator cmd now.)
* Verify there is no OUTPUT.txt in C:\TEST
* Run the winzip installation msi from SYSTEM cmd.
* Check if OUTPUT.txt appears in C:\TEST.
* Inspect the OUTPUT.txt. **NT AUTHORITY\SYSTEM** , means our test64.exe ran as a SYSTEM user => **CONCLUSION** the installation is vulnerable.

Notice: Race conditions are more difficult to exploit. It could require multiple reruns of the installation to demonstrate exploitation.