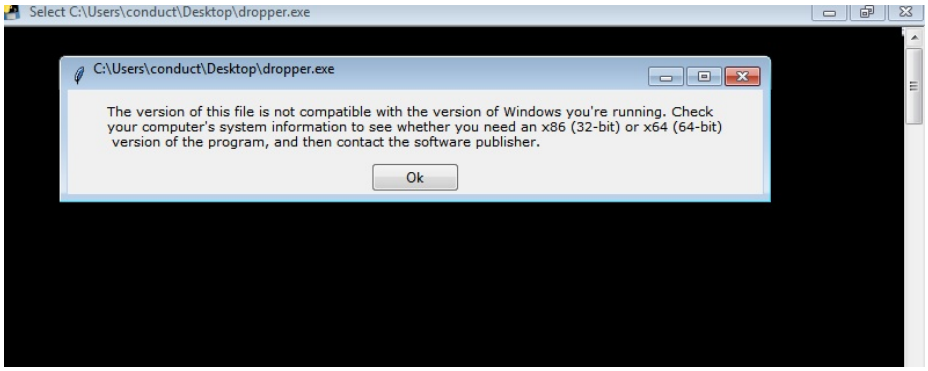
Dynamic Analysis of PySock + Dropper

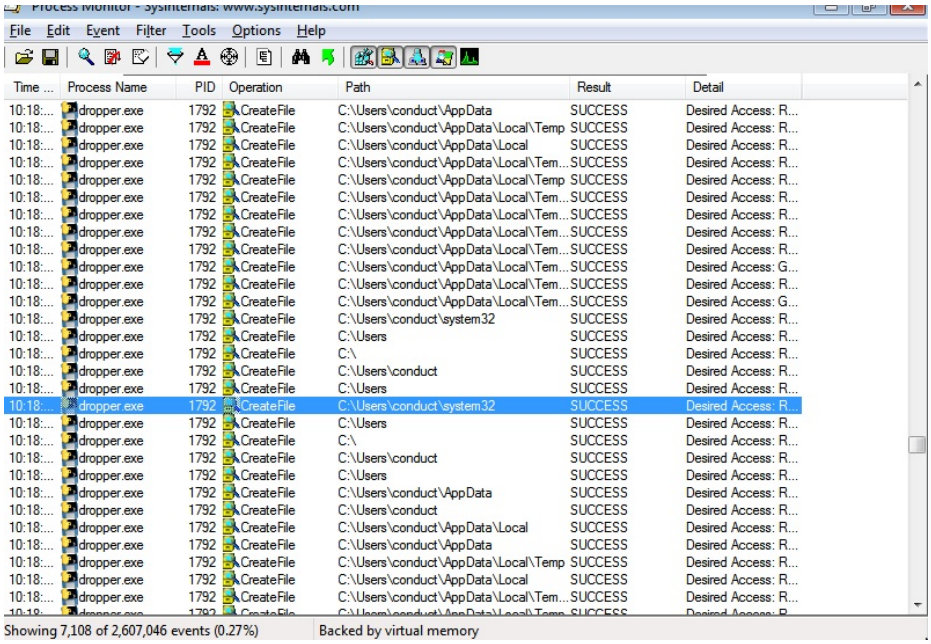
For this analysis, we will be skipping the part where the reverse shell lands on the victim's system.

One thing to note is that if you compile the python file on a Windows 7 machine then attempt to execute it on a Windows 10 machine, you will run into issues. But if you compile it on a Windows 10 machine then attempt to execute it on an older Windows machine, it will run successfully.

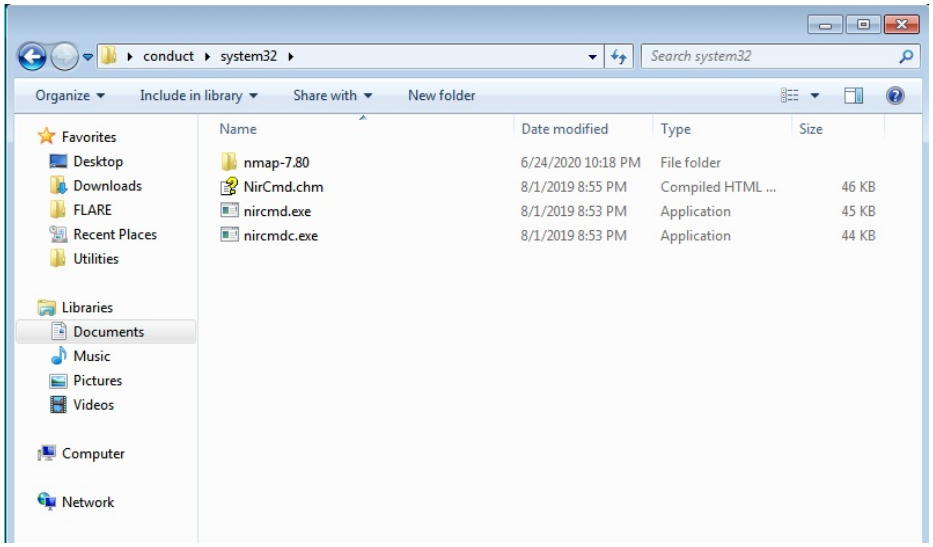
After a few seconds of executing the dropper.exe file, we get this popup. Looking at the further left-hand corner, we see a feather which tells us it isn't an actual Windows error message, moreso it's a message coded in to make the victim think the "software" didn't work & cause them to delete the software.



Looking at Procmon after running the dropper.exe file, we see a ton of events relating to file creation that the file executes. A few that stand out mention a system32 directory in %USERPROFILE%:

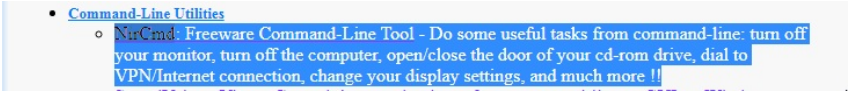


After navigating into the %USERPROFILE%\system32 directory, we see some new entries:

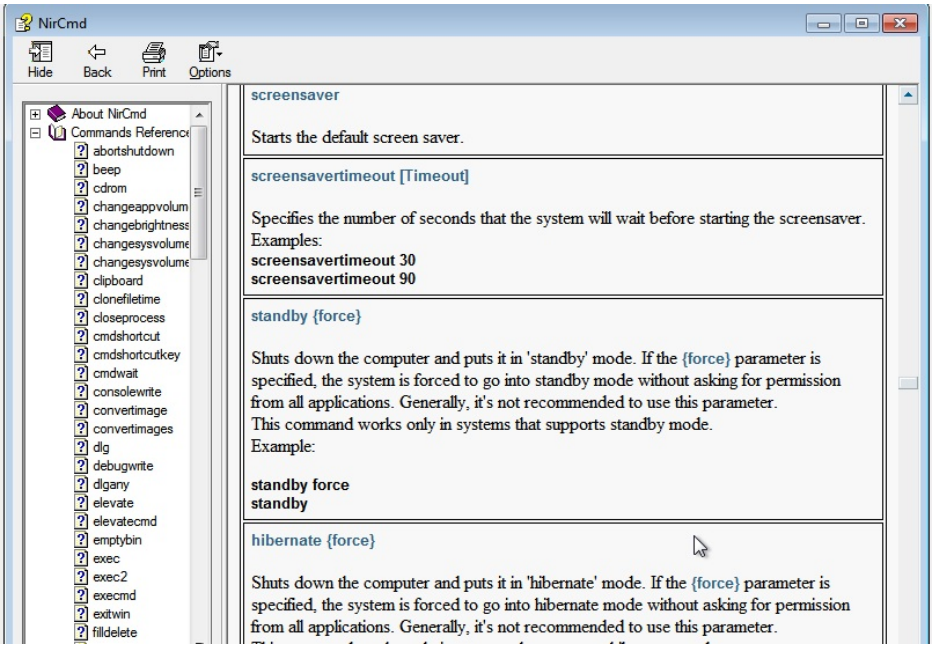


The dropper.exe file drops an nmap folder as well as 3 files relating to nircmd.

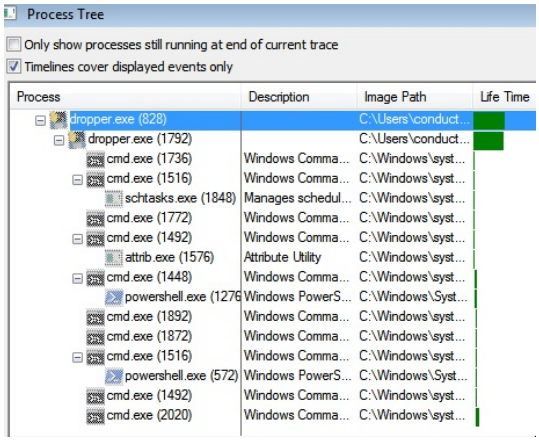
If we go into the command line & execute the nircmd executable, we get a popup saying to go to a website for more info, upon doing that we see that nircmd is a command line utility that has many functions:



The nircmd.CHM file is like a man page, it shows all the functionality of the executable:

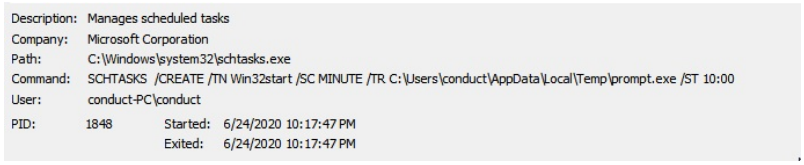


Looking at the process tree section of Procmon, we get a better view of the events occurring after executing the dropper.exe.



As we can see, right after executing it, schtasks.exe is executed which tells us that a new task was scheduled. We also see attrib.exe being executed which is a program used to add attributes such as making a file read-only or hidden. And the last thing we see is powershell being executed on 2 separate occassions in this tree.

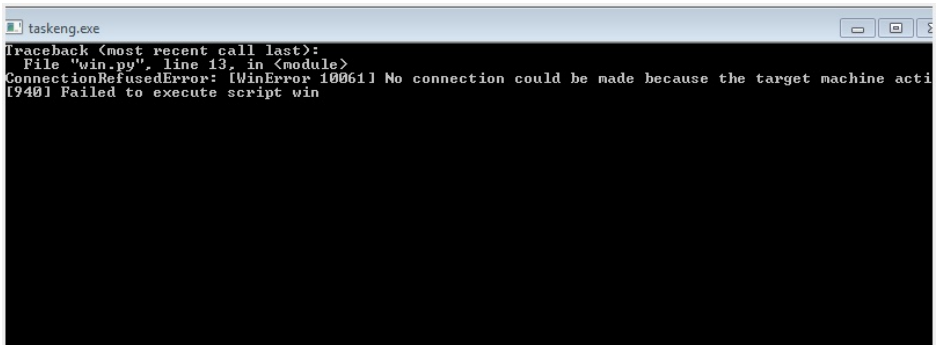
Going more in-depth into the scheduled task with process tree:



We can see the exact command used to create the task was:

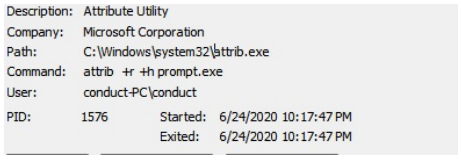
`SCHTASKS /CREATE /TN Win32start /SC MINUTE /TR C:\Users\conduct\Appdata\TEMP\prompt.exe /ST 10:00`
This creates a task named Win32start & schedules it to execute the prompt.exe file in TEMP every minute, starting at 10:00 AM.

Occassionally, the victim's machine will flash with a taskeng.exe command prompt, this is the scheduled task attempting to run.



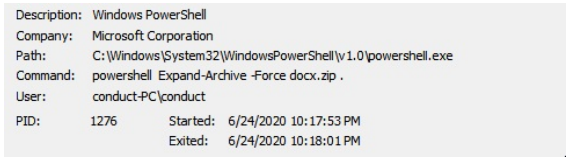
Here, we see that the task failed to run & errors out with a connection refused, hinting that the prompt.exe is attempting to connect back to the attacker.

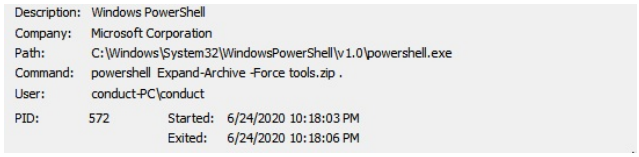
Looking at the attrib.exe event, we see that prompt.exe acquires the read only (+r) and hidden (+h) attributes.



I believe that adding the read only attribute prevents the user from deleting the file:

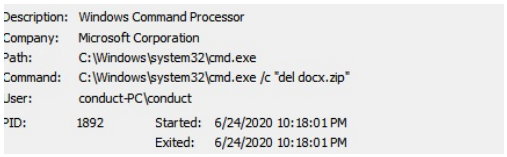
Moving on to the two Powershell events, we see that their commands are similar:

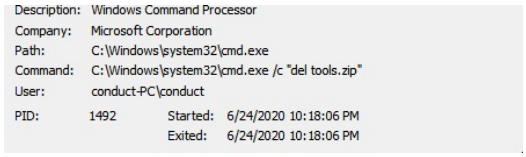




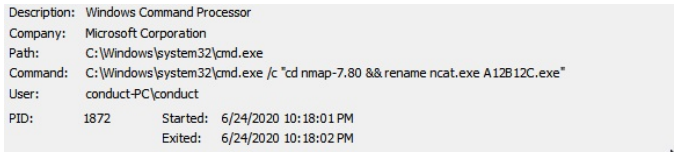
Both commands are unzipping files, one zip file being named docx while the other is named tools

And right after each powershell command is executed , the zip files are deleted:





Another interesting command I stumbled upon was the renaming of a file inside the nmap folder we seen earlier:

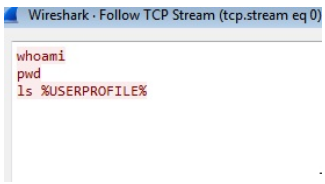


Here, the ncat file is being renamed to A12B12C12.exe i order to avoid detection when it's being used.

Network Traffic

Here, we set up an attacker machine & use netcat to listen for the victim's machine to connect back to us.

We start sending commands to the victim's machine & look at the traffic via Wireshark. The blue is the victim's response while the red is the attacker's requests:

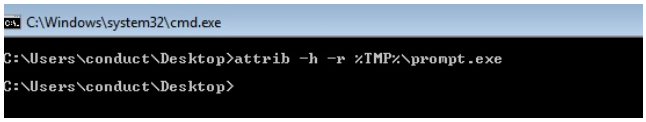


```
conduct-pc\conduct
C:\Windows\system32
AppData
Application Data
Contacts
Cookies
Desktop
Documents
Downloads
Favorites
Links
Local Settings
Music
My Documents
NTUSER.DAT
NTUSER.DAT{016888bd-6c6f-11de-8d1d-001e0bcde3ec}.TM.blf
NTUSER.DAT{016888bd-6c6f-11de-8d1d-001e0bcde3ec}.TMContainer00000000000000
NTUSER.DAT{016888bd-6c6f-11de-8d1d-001e0bcde3ec}.TMContainer00000000000000
NetHood
Pictures
PrintHood
Recent
Saved Games
Searches
SendTo
Start Menu
Templates
```

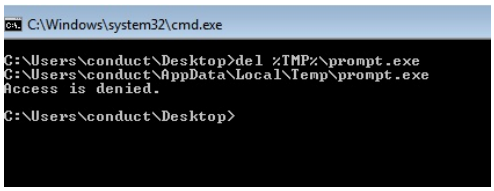
Removal

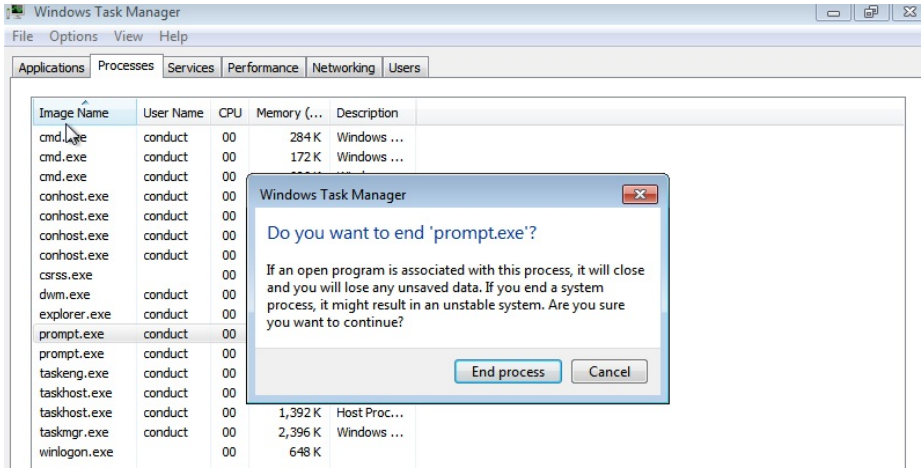
Removing this malware is fairly straightforward.

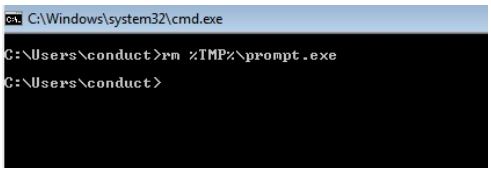
- Remove the read-only & hidden attribute from prompt.exe



- Delete prompt.exe. **NOTE**: If there is an active connection to the attacker, you won't be able to remove the prompt.exe file, you're going to have to kill the process then remove the file:







- Finally, remove teh dropper.exe file and you're Pysock free!