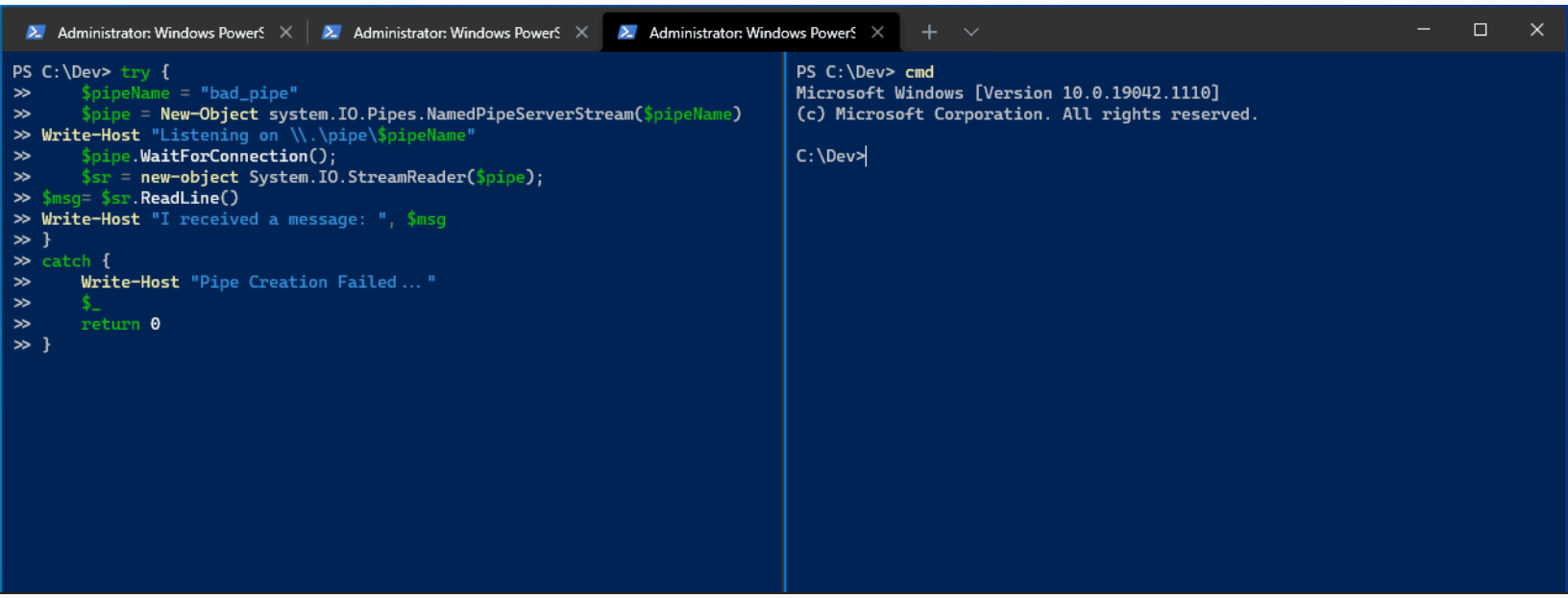
Guide to Named Pipes and Hunting for Cobalt Strike Pipes

```
try {
 $pipeName = "bad_pipe"
 $pipe = New-Object
system.IO.Pipes.NamedPipeServerStream($pipeName)
 Write-Host "Listening on \\.\pipe\$pipeName"
 $pipe.WaitForConnection();
 $sr = new-object System.IO.StreamReader($pipe);
 $msg= $sr.ReadLine()
 Write-Host "I received a message: ", $msg
}
catch {
 Write-Host "Pipe Creation Failed..."
 $_
 return 0
}
```



blog.cobaltstrike.com

In this blog, Raphael Mudge (the creator of Cobalt Strike), notes some of the default pipe names. You can also customise the names of these pipes using Malleable C2 profiles.

See a sample of regexes for pipe names I put together from default and custom profiles below:

```
LET pipeRegex = 'bad_pipe'
LET processes = SELECT Pid AS ProcPid, Name AS ProcName, Exe
FROM pslist()
WHERE ProcPid > 0

SELECT * FROM foreach(
row=processes,
query={
    SELECT ProcPid, ProcName, Exe, Type, Name, Handle
    FROM handles(pid=ProcPid)
    WHERE Name =~ pipeRegex
})
```

In Cobalt Strike, the interface for creating a new SMB listener the default pipe name was `msagent_f8` which matches what we learnt before. I ran `jump psexec_psh` to laterally move to a different host.

This was a good start and found named pipes such as the SMB beacon that stay open for a long period of time, but it doesn't catch the transient named pipes.

Of course, if you are lucky enough to have Sysmon deployed to the network already, you can easily monitor for these same named pipes as shown below:

Written by svch0st

317 Followers

Follow