# Guide to Named Pipes and Hunting for Cobalt Strike Pipes

svch0st · Follow
4 min read · Jul 25, 2021



## Intro to Named Pipes

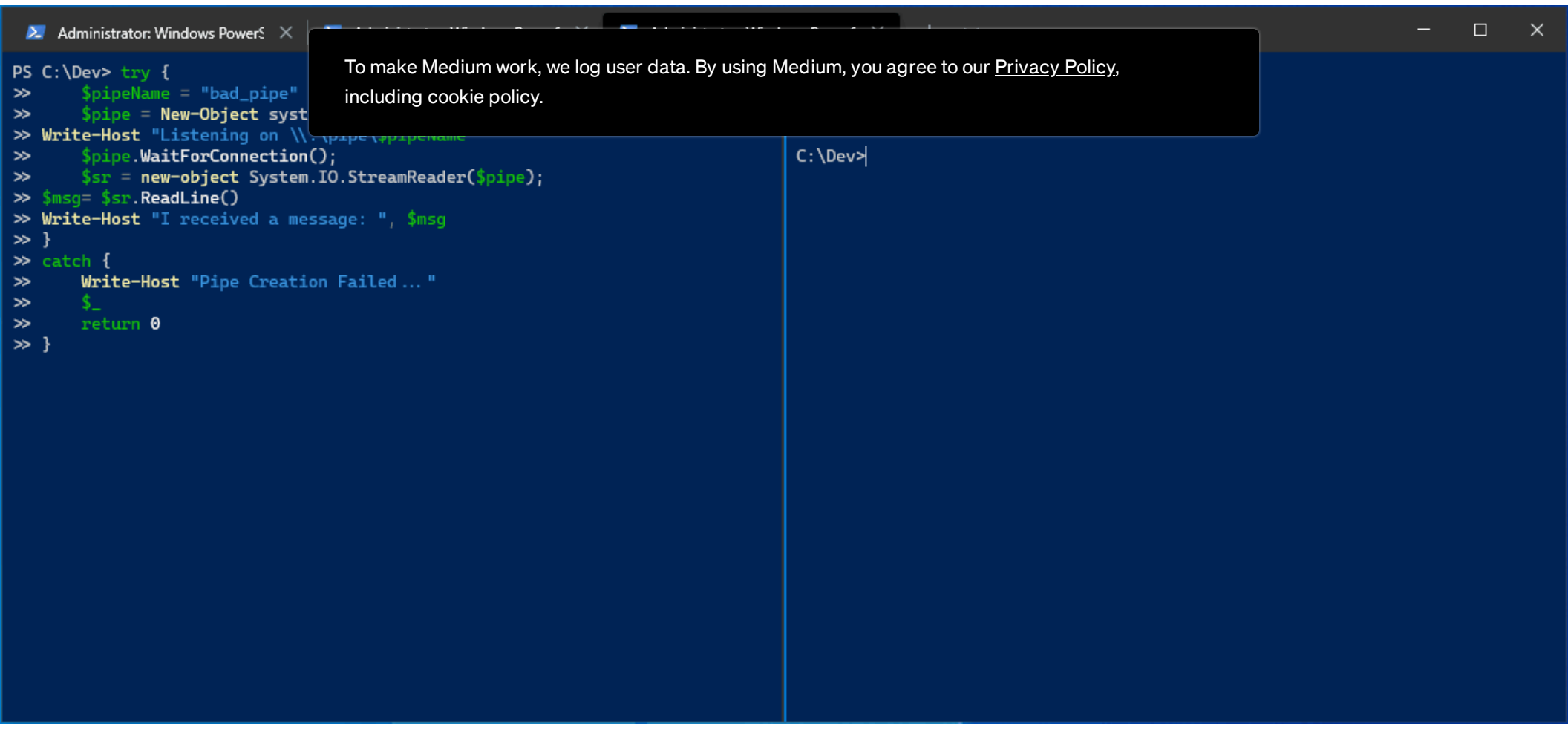
The way that helped me start to understand pipes is to think of them as like type of network socket that is created. It can be used to send and receive information between processes or even hosts.

As a rudimentary example, you can query the current pipes on your host:

```
Get-ChildItem \\.\pipe\
```

Now lets try creating one. Below is a basic script to create a named pipe using PowerShell:

```
try {
```





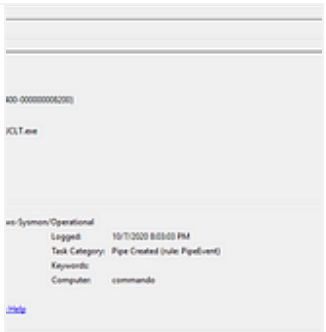

## How Cobalt Strike uses Named Pipes

There is heaps of existing research on how Cobalt Strike utilises named pipes:

**Detecting Cobalt Strike Default Modules via Named Pipe Analysis**

During recent years, the Cobalt Strike framework has gained significant popularity amongst red teamers and threat...

labs.f-secure.com



### Including from the Cobalt Strike blog:

## Using Velociraptor to Search for Malicious Named Pipes

When a process uses a named pipe, it creates a handle. Below is a sample of VQL that will walk through all running processes and pull the handles of the process. It will then search for any handles that match the regex `bad_pipe`.

It recorded what process was using the pipe as well as the pipe name! Using the regex of some of the default named pipes lets put all this to the test.

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.

Th

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:

| i | _time | host | TaskCategory | EventCode | Image | PipeName | ProcessId |
|---|---|---|---|---|---|---|---|
| > | 7/24/21 3:08:29.000 PM | dc.windomain.local | Pipe Created (rule: PipeEvent) | 17 | C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe | \msagent_f8 | 3620 |
| > | 7/24/21 3:08:28.000 PM | dc.windomain.local | Pipe Created (rule: PipeEvent) | 17 | C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe | \status_8f | 3620 |

Edit: I'm currently researching the possibility of monitoring named pipes with ETW and using Velociraptor further.

Thanks,

@svch0st

Cobalt Strike Dfir

svch0st

**Event Log Tampering Part 2: Manipulating Individual Event Logs**

This is Part 2 in a look at Event Log Tampering. Check out Part 1 before reading...

Oct 2, 2020 65 1

svch0st

**Windows User Access Logs (UAL)**

Overview

Feb 25, 2021 69

svch0st

**Event Log Tampering Part 1: Disrupting the EventLog Service**

Windows event logs are a fundamental source of data and evidence for incident response....

Oct 1, 2020 21

svch0st

**Active Directory Recon Cheat-sheet**

Find Goodies

Jul 3, 2019 18

## Lists

**Staff Picks**
755 stories · 1416 saves

**Stories to Help You Level-Up at Work**
19 stories · 852 saves

**Self-Improvement 101**
20 stories · 2960 saves

**Productivity 101**
20 stories · 2506 saves

Aardvark Infinity in Aardvark Infinity

**Set Up a Windows 11 Malware Analysis Lab for Reverse...**

Aug 29 12

Shreyash Tambe

**Cracking AgentTesla: Revealing Steganographic Payload**

Reversing .NET Executables and DLL Payloads

Sep 5 4