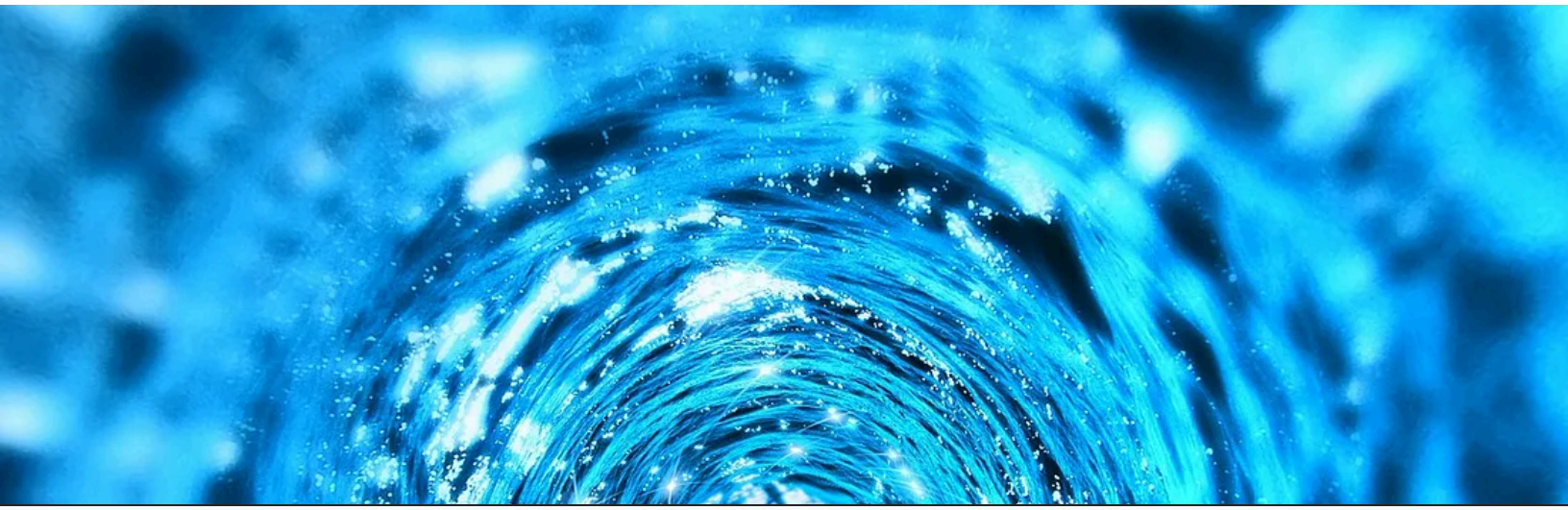
Detecting ‘Dev Tunnels’

BlueteamOps · Follow

Published in Detect FYI · 5 min read · Oct 23, 2023



of multiple different ports using the same tunnel. It leverages Microsoft's Dev Tunnel relay service (a cloud service) which secure communications between a dev tunnel host and clients..

Adversaries can abuse these tools for command and control, persistence, and to evade defenses. There are already red team writeups available on how Dev Tunnels can be being used for C2. Hence, I thought of spending sometime using this tool to get to know it better so defenders can be ready. In this article I will be going through detection opportunities when Dev Tunnels is being used for tunnelling RDP

. . .

Detection Opportunities

Network detections

Following DNS queries were recorded in my test hosts and the queries occurred in the following order.

DNS queries (clusterId's highlighted)

Before creating a tunnel, a user must log in with either a Microsoft ID or GitHub ID. The command line arguments are as follows:

```
Authenticates using Microsoft

devtunnel user login

Authenticates using Github with either browser based or via device code

devtunnel user login -g
devtunnel user login -d
```

```
devtunnel.exe host -p 3389 - allow-anonymous
```

A persistent tunnel with anonymous access targeting port 3389 can be created by running the following command sequence.

```
devtunnel.exe create -a
devtunnel.exe port create -p 3389
```

Below list contains the most notable files since Dev tunnels leverages SSH to carryout secure communications and port forwarding activities.

Microsoft.DevTunnels.Ssh.Tcp.dll
Microsoft.DevTunnels.Ssh.dll
Microsoft.DevTunnels.Management.dll
Microsoft.DevTunnels.Contracts.dll
Microsoft.DevTunnels.Connections.dll

Following registry set activity was recorded when the devtunnel.exe process

Event ID 25 and Event ID 24 from Microsoft-Windows-TerminalServices-LocalSessionManager/Operational contains the Source Network Address with the value **%16777216** which is a positive indication of tunnelling activity (not just applicable to Dev Tunnels but others as well). Sophos and Logpoint articles also covers this in detail.

Figure 5 and Figure 6 shows successful authentication through the tunnel. Initial 4624 event had the Logon type as 3 and also leaks the adversary workstation name. The source address is shown as ::1 and source port as 0. Within seconds we see the Logon type 10 record with the same source address and port. However, in this instance the workstation name has the value of the host that runs the tunnel and not the hostname of adversary's machine.

Detection Engineering | Threat Detection | Threat Hunting | Threat Research | Dfir