Powercat

PowerShell for Pentester

Table of Contents

Abstract

Powercat is a simple network utility used to perform low-level network communication operations. The tool is an implementation of the well-known Netcat in Powershell. Traditional anti-viruses are known to allow Powercat to execute.

The installed size of the utility is 68 KB. The portability and platform independence of the tool makes it an essential arrow in every red teamer's quiver. In this report, we'll demonstrate and learn the functionality of this tool.

Disclaimer: This report is provided for educational and informational purpose only (Penetration Testing). Penetration Testing refers to legal intrusion tests that aim to identify vulnerabilities and improve cybersecurity, rather than for malicious purposes.

Introduction

Powercat is a program that offers Netcat's abilities to all current versions of Microsoft Windows. It tends to make use of native PowerShell version 2 components.

We need to go to the website listed in the section of references. Users may download the link because it is a Github website.

Basic Options in Powercat

Powercat supports various options to play around with. We'll cover the following in this article.

-l	Listen for a connection
-c	Connect to a listener
-p	The port to connect to, or listen on
-e	Execute
-ep	Execute Powershell
-g	Generate Payload
-ge	Generate Encoded Payload
-d	Disconnect stream
-i	Input data

Setting up Powercat

Powershell execution policy is a safety feature in Windows which determines which scripts can or cannot run on the system, therefore, we need to set the Powershell execution policy to "bypass." This would allow all scripts to run without restriction. Thereafter, we need to download Powercat using wget.

```
powershell -ep bypass
```

```
wget https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1 -o powercat.ps1
```

```
PS C:\Users\ignite\Desktop> powershell -ep bypass
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\ignite\Desktop> wget https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1 -o powercat.ps1
PS C:\Users\ignite\Desktop> ls


    Directory: C:\Users\ignite\Desktop


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        10/13/2021   9:43 AM          37667 powercat.ps1
```

Now that we have downloaded the Powercat script, we can import it into the current Powershell terminal and then it could be used.

```
Import-Module .\powercat.ps1
```

```
PS C:\Users\ignite\Desktop> Import-Module .\powercat.ps1
PS C:\Users\ignite\Desktop> powercat -h

powercat - Netcat, The Powershell Version
Github Repository: https://github.com/besimorhino/powercat

This script attempts to implement the features of netcat in a powershell
script. It also contains extra features such as built-in relays, execute
powershell, and a dnscat2 client.

Usage: powercat [-c or -l] [-p port] [options]

  -c  <ip>          Client Mode. Provide the IP of the system you wish to connect to.
                    If you are using -dns, specify the DNS Server to send queries to.

  -l                Listen Mode. Start a listener on the port specified by -p.

  -p  <port>        Port. The port to connect to, or the port to listen on.

  -e  <proc>        Execute. Specify the name of the process to start.

  -ep               Execute Powershell. Start a pseudo powershell session. You can
                    declare variables and execute commands, but if you try to enter
                    another shell (nslookup, netsh, cmd, etc.) the shell will hang.

  -r  <str>         Relay. Used for relaying network traffic between two nodes.
                    Client Relay Format:   -r <protocol>:<ip addr>:<port>
```

Port Scanning

Powercat is equipped with the functionality to scan for open ports. It is able to do this by attempting a TCP connection to the ports defined. For example, if I have to check for a running service on port 21,22,80,443, we can do this by:

```
(21,22,80,443) | % {powercat -c 192.168.1.150 -p $_ -t 1 -Verbose -d}
```