Telstra

May 24th 2018

E2ESA PingScript

VERSION 1.0 DOCUMENTATION

End-2-End Service Analysis

Customer Service Improvement | Customer Service Management | Telstra Operations

For support, contact:

Jeremy.Coombs@team.telstra.com

Table of Contents

Context:

The need for this tool was identified by the End-to-End Service Analysis (E2ESA) team during the 2017 Siebel E2E Network Performance project.

The project involved gathering network statistics from remote contact centres (onshore and offshore) to assess baseline performance.

To gather this data, a network baselining script was developed to measure network latency and packet loss from an agent's PC to any other specified endpoint. Initially this was a batch script developed by Ivan Liang that was then interpreted and plotted to a graph by a Python script written by Kim Alford.

Over time, this script has been modified and additional functionality added so that it could be applied in different scenarios.
The aim in mind has been to reduce the interaction required by the user.

Purpose:

This script is used to measuring the network performance at some remote location that is inaccessible over the EDN because of firewalls or other network boundaries. It is a good way to understand the performance of the network as experienced from that remote location.

It may also help in determining the source of any network issues being experienced. For example, if you know that there is a lot of packet loss occurring somewhere between two points in the network, this tool can help to determine which node in a path is causing packet loss.

This script was written and packaged with ease of use in mind, so that there is no need to rely on technical skill of whoever is running it. It is possible to run an entire series of tests lasting many hours by using a single command. This, as well as the different use-cases of this tool will be covered in the "Instructions" section.

Accessing the Tool:

This tool can be downloaded as a .zip archive from the E2ESA webserver, currently at this address:
http://e2esa-test-server/baseline/files/E2ESA_PingScript.zip
Within this archive is the executable file (.exe), the original Python file (.py), and this documentation explaining how the tool is to be used.

It can also be found on Telstra's GitLab site here:
https://git.puma.corp.telstra.com/d862217/ping_script
This requires you to sign up for the free LDAP authenticated account.

A more permanent and accessible home is being reviewed.

Instructions:

This is a command line (CLI) tool that can be run in two different modes:

a) **Command Line Arguments**
   It can be run with all necessary options passed to the program as a single command line string. Once run, this method requires no further interaction from the user.

b) **Interactive**
   Running in this mode, the user will interact with the script via a menu that accepts input.

When using either mode, the tests can be performed in 3 different ways.
These correspond to the menu options, and to the command line options:

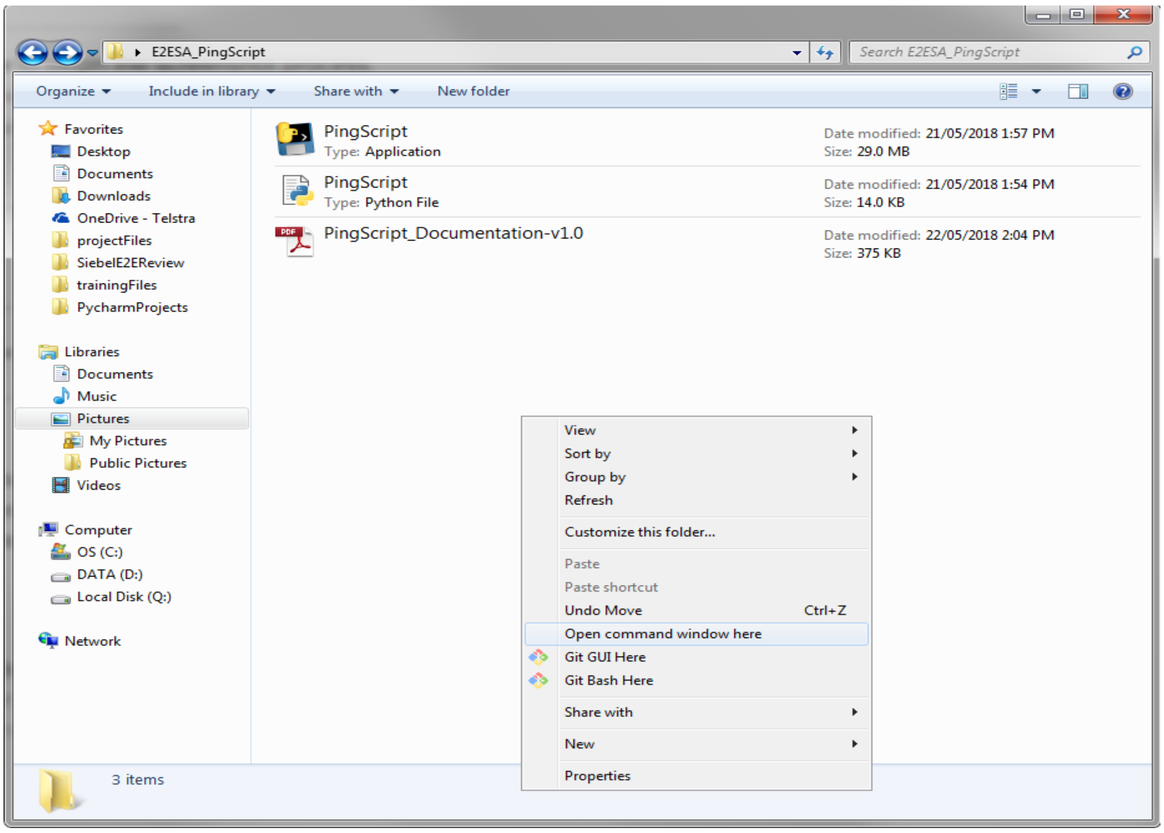
1) Test a single specified host,
2) Test a number of specified hosts,
   - If a file named '*hosts.txt*' is found in the directory, the hosts will be taken from this.
3) Ping each node in the route to any specified host.

Once the tests have completed, all the relevant files will be archived into a .zip file to make it easier for these files to be sent back to Telstra analysts via email or file transfer.

Mode #1: Command Line Arguments

The script can accept command line arguments at the time that it is run. This allows the user to run the script, and to enter all the information that the script will need to run in a single command line string.

To run the script in this mode, open a command prompt (CMD.exe) window in the directory containing the executable version of the script.
The easiest way to do this is to hold down *Shift* and *right-click* in the directory, then select "*Open command window here*" from the menu. See the image below:



PingScript
Type: Application
Date modified: 21/05/2018 1:57 PM
Size: 29.0 MB

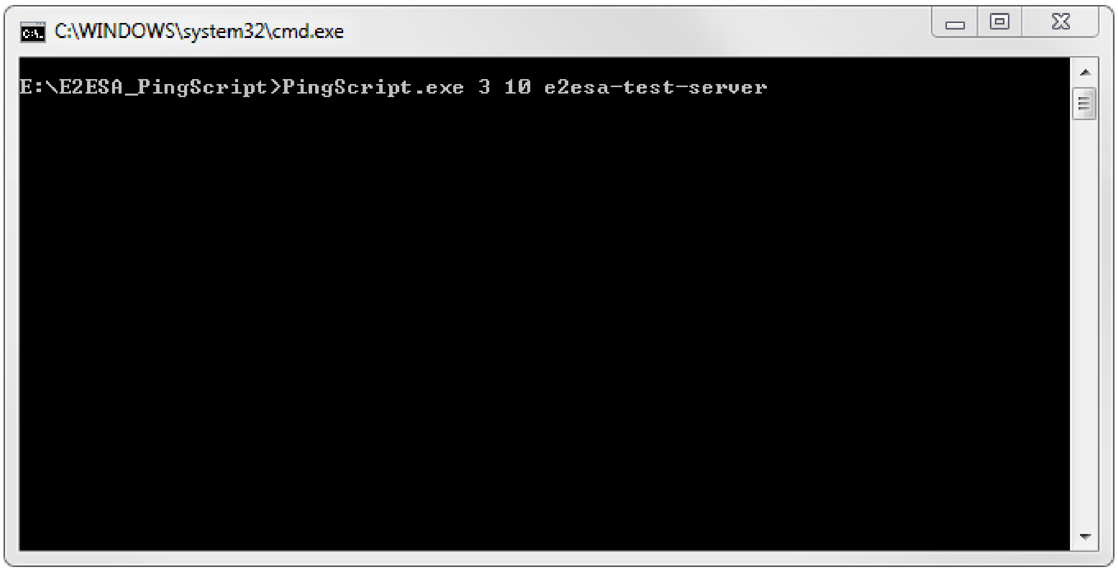
PingScript
Type: Python File
Date modified: 21/05/2018 1:54 PM
Size: 14.0 KB

PingScript_Documentation-v1.0
Date modified: 22/05/2018 2:04 PM
Size: 375 KB

Alternatively, you could open the CMD window anywhere and navigate to the directory if you know how.

The structure of the command line string is as follows:
ProgramName   TestOption   Duration (in minutes)   Host#1   Host#2   HostEtc.

The example below will run option #3 of the script (Ping each node in the route to any specified host), for 10 minutes each test, and with the destination host *e2esa-test-server*.



C:\WINDOWS\system32\cmd.exe

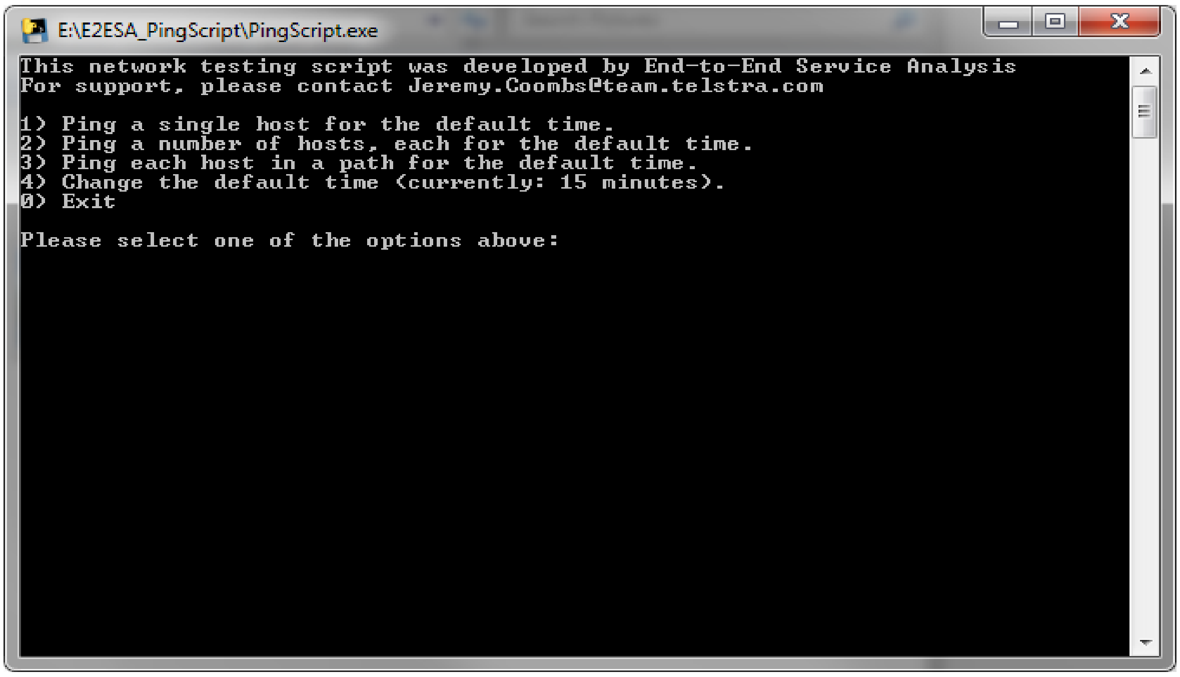
```
E:\E2ESA_PingScript>PingScript.exe 3 10 e2esa-test-server
```

After pressing *Enter*, the script will test each node for 10 minutes and save all of the files to a separate subdirectory. There is no need for further input from the user.

Mode #2: Interactive Menu

This tool can also be run like any other program, by double-clicking on the executable file.

After doing this, the user will be greeted by a menu which requires the user to select one of the options provided:



E:\E2ESA_PingScript\PingScript.exe

```
This network testing script was developed by End-to-End Service Analysis
For support, please contact Jeremy.Coombs@team.telstra.com

1) Ping a single host for the default time.
2) Ping a number of hosts, each for the default time.
3) Ping each host in a path for the default time.
4) Change the default time (currently: 15 minutes).
0) Exit

Please select one of the options above:
```

The first 3 options correspond to the 3 different ways that the test(s) can be run, as outlined above. By selecting option ‘4’, the user can change the time that each test will be run. This defaults to 15 minutes per test.

When selecting one of the 3 testing methods, the user will then be prompted to enter the hostname(s) that they wish to test.

Note: If option 2 is selected, the script will first look for a file named 'hosts.txt' before prompting the user to enter the hostnames manually.

After this point, no further interaction is required from the user.

How It Works:

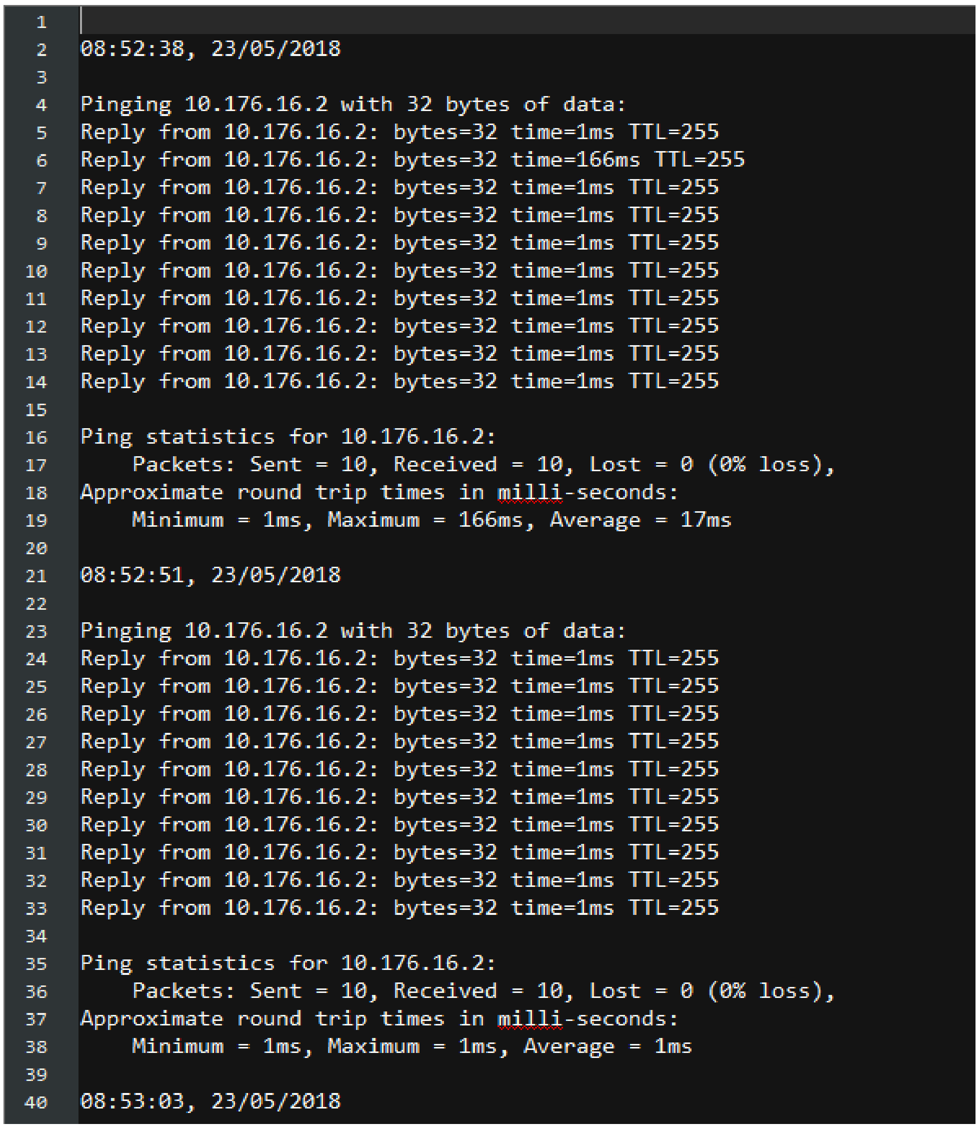
At a high level, the script works by recording the output of a series of pings with time stamps to a file, then reading through the file to gather average latency associated with each time stamp and plotting these results to a graph. Finally, the script packages all the relevant files into a separate subdirectory, and then compresses this directory into a .zip archive.

See breakdown below for a more detailed view:

1) Append date-time stamp to {hostname}.log file,
   - datetime.now().strftime("%H:%M:%S, %d/%m/%Y")
2) Append ping results to {hostname}.log file in blocks of 10,
   - ping –n 10 *hostname*
3) Read through {hostname}.log file and save variables to lists:
   - append date-times to list(x)
   - append latency to list(y)
   - where packet is dropped, append to list(xt) and list(yt)
4) Plot lists onto graph using matplotlib.pyplot
   - list(x) and list(y) plot average latency over time as blue line,
   - list(xt) and list(yt) plot lost packets as red dot.
5) Save graph as {}hostname}.png image,
6) Create subdirectory with unique name:
   - {PC_hostname}_datetime.now().strftime("%d%m%Y_(%H.%M)")
   - Example: *W00000PC0N0RX8_23052018_(09.09)*
7) Move all .txt, .log, and .png files to subdirectory created above,
8) Create .zip archive of above subdirectory,
9) End.

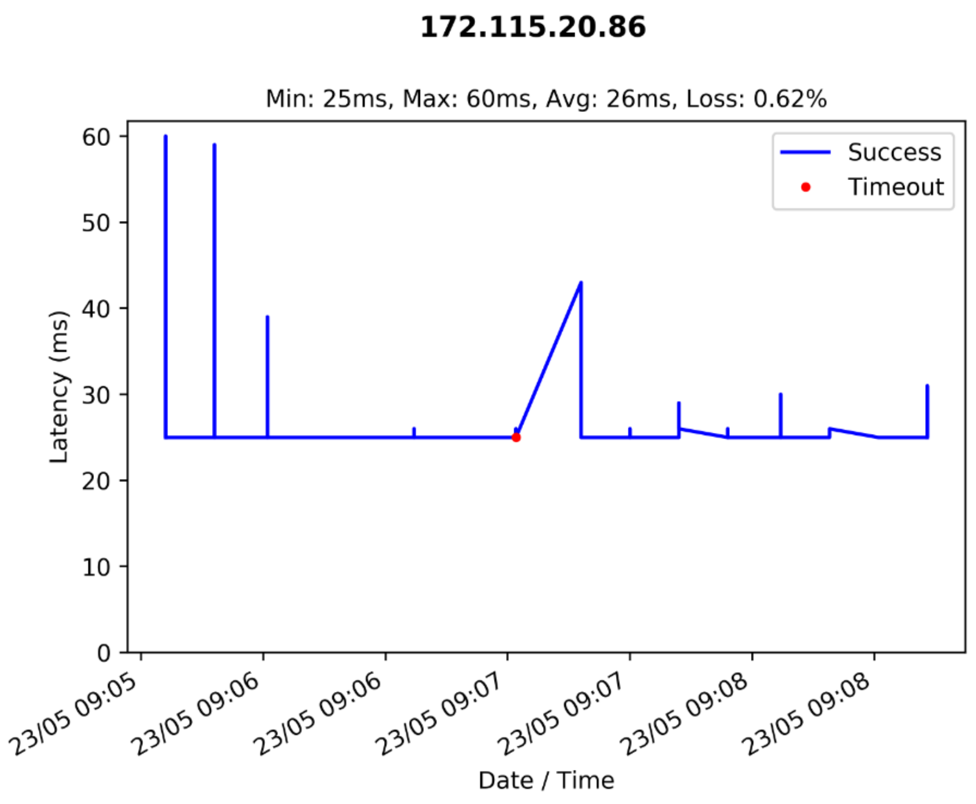
Examples:

{*hostname*}.log file:

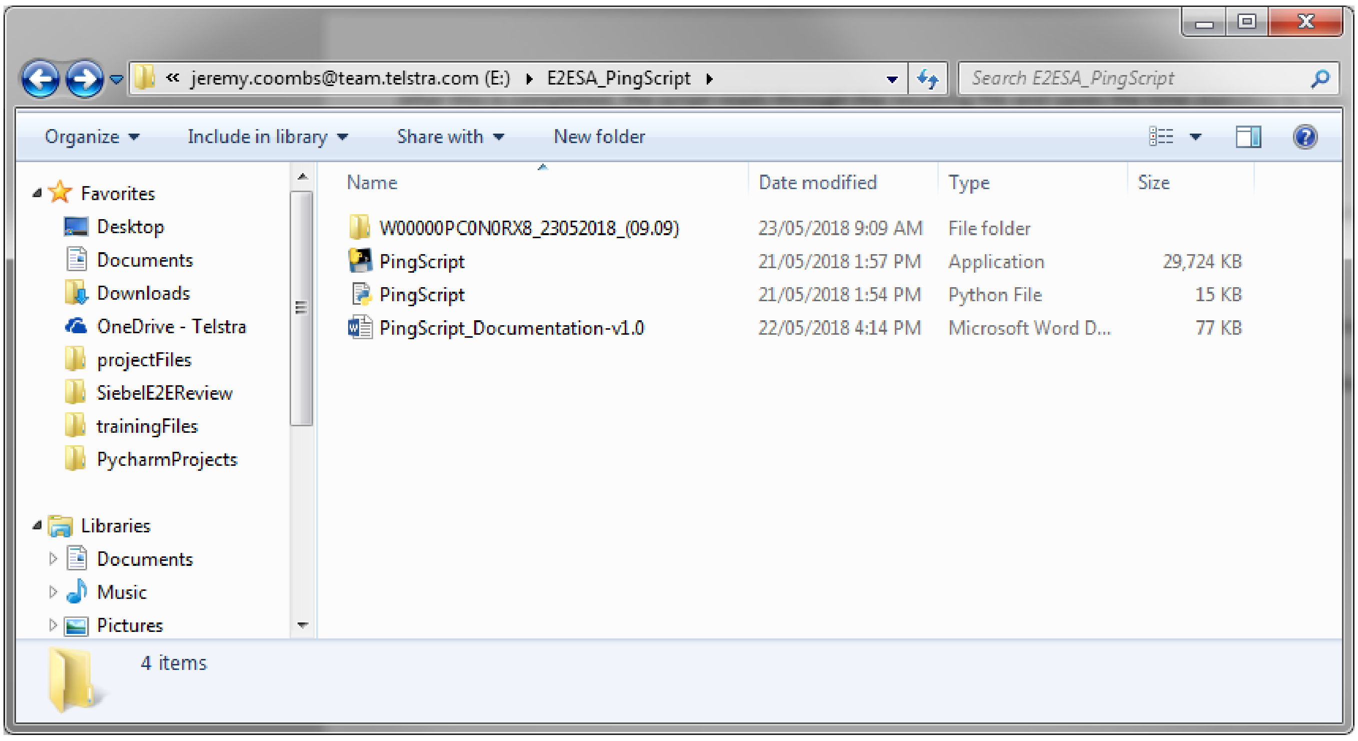


```

08:52:38, 23/05/2018

Pinging 10.176.16.2 with 32 bytes of data:
Reply from 10.176.16.2: bytes=32 time=1ms TTL=255
Reply from 10.176.16.2: bytes=32 time=166ms TTL=255
Reply from 10.176.16.2: bytes=32 time=1ms TTL=255
Reply from 10.176.16.2: bytes=32 time=1ms TTL=255
Reply from 10.176.16.2: bytes=32 time=1ms TTL=255
Reply from 10.176.16.2: bytes=32 time=1ms TTL=255
Reply from 10.176.16.2: bytes=32 time=1ms TTL=255
Reply from 10.176.16.2: bytes=32 time=1ms TTL=255
Reply from 10.176.16.2: bytes=32 time=1ms TTL=255
Reply from 10.176.16.2: bytes=32 time=1ms TTL=255

Ping statistics for 10.176.16.2:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 166ms, Average = 17ms

08:52:51, 23/05/2018

Pinging 10.176.16.2 with 32 bytes of data:
Reply from 10.176.16.2: bytes=32 time=1ms TTL=255
Reply from 10.176.16.2: bytes=32 time=1ms TTL=255
Reply from 10.176.16.2: bytes=32 time=1ms TTL=255
Reply from 10.176.16.2: bytes=32 time=1ms TTL=255
Reply from 10.176.16.2: bytes=32 time=1ms TTL=255
Reply from 10.176.16.2: bytes=32 time=1ms TTL=255
Reply from 10.176.16.2: bytes=32 time=1ms TTL=255
Reply from 10.176.16.2: bytes=32 time=1ms TTL=255
Reply from 10.176.16.2: bytes=32 time=1ms TTL=255
Reply from 10.176.16.2: bytes=32 time=1ms TTL=255

Ping statistics for 10.176.16.2:
    Packets: Sent = 10, Received = 10, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 1ms, Average = 1ms

08:53:03, 23/05/2018
```

{*hostname*}.png file:

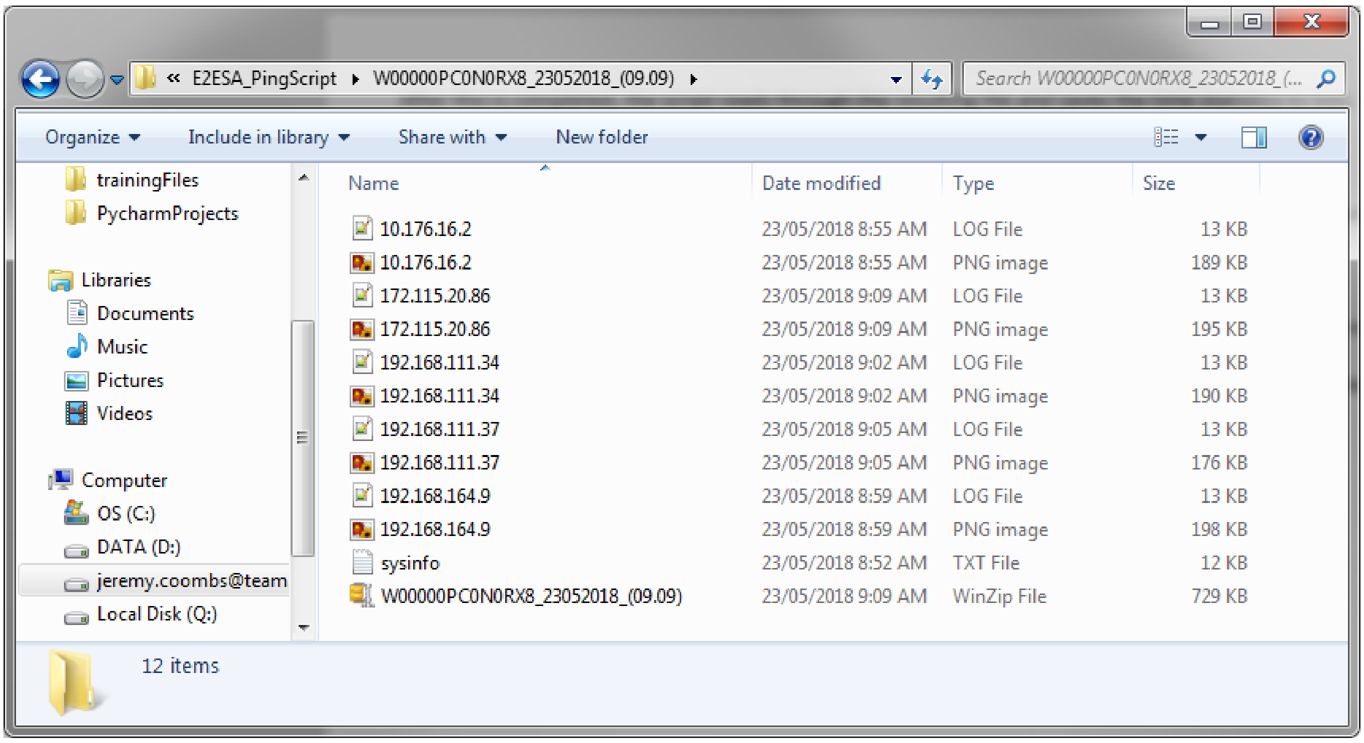


Subdirectory Creation:



Name	Date modified	Type	Size
W00000PC0N0RX8_23052018_(09.09)	23/05/2018 9:09 AM	File folder	
PingScript	21/05/2018 1:57 PM	Application	29,724 KB
PingScript	21/05/2018 1:54 PM	Python File	15 KB
PingScript_Documentation-v1.0	22/05/2018 4:14 PM	Microsoft Word D...	77 KB

Subdirectory Contents:



« E2ESA_PingScript ▸ W00000PC0N0RX8_23052018_(09.09) ▸

Search W00000PC0N0RX8_23052018_(...

Name	Date modified	Type	Size
10.176.16.2	23/05/2018 8:55 AM	LOG File	13 KB
10.176.16.2	23/05/2018 8:55 AM	PNG image	189 KB
172.115.20.86	23/05/2018 9:09 AM	LOG File	13 KB
172.115.20.86	23/05/2018 9:09 AM	PNG image	195 KB
192.168.111.34	23/05/2018 9:02 AM	LOG File	13 KB
192.168.111.34	23/05/2018 9:02 AM	PNG image	190 KB
192.168.111.37	23/05/2018 9:05 AM	LOG File	13 KB
192.168.111.37	23/05/2018 9:05 AM	PNG image	176 KB
192.168.164.9	23/05/2018 8:59 AM	LOG File	13 KB
192.168.164.9	23/05/2018 8:59 AM	PNG image	198 KB
sysinfo	23/05/2018 8:52 AM	TXT File	12 KB
W00000PC0N0RX8_23052018_(09.09)	23/05/2018 9:09 AM	WinZip File	729 KB

12 items