# Network Service Test Scripts

## Introduction

This repository contains two scripts (Python and PowerShell) designed to help network and system administrators test basic network connectivity and service availability for a list of target hosts. The scripts can check reachability using Ping (ICMP) and test if HTTP and HTTPS services are responding correctly.

Results are displayed in the console with color-coded status indicators, and can optionally be exported to a CSV file for logging and analysis.

**Current Date:** Saturday, April 5, 2025

## Features

* Test **Ping** (ICMP echo request/reply).
* Test **HTTP** (TCP port 80) connectivity and basic response.
* Test **HTTPS** (TCP port 443) connectivity and basic response (with SSL/TLS).
* Input targets via:
* Command-line arguments (for a single host).
* A CSV file (for multiple hosts).
* **Colored console output** for easy identification of SUCCESS (Green), FAILED (Red), and SKIPPED/WARNING (Yellow) statuses.
* Optional **CSV export** of detailed test results including timestamp, host, service, status, and details.
* Available in both **Python** and **PowerShell**.

## Requirements

### Python Version (`network\_test.py`)

* **Python:** 3.6 or higher recommended.
* **Libraries:** requests, colorama
* Install using pip:

pip install requests colorama

### PowerShell Version (`network\_test.ps1`)

* **PowerShell:**
* Windows PowerShell 5.1 (comes with Windows 10/Server 2016+)
* PowerShell Core 7.x+ (recommended, cross-platform)
* **Execution Policy:** You may need to adjust your PowerShell execution policy to run local scripts. To allow signed scripts (or all local scripts), run PowerShell as Administrator and execute:

# Check current policy (optional)

Get-ExecutionPolicy -List

# Allow running local scripts for the current user (often sufficient)

Set-ExecutionPolicy RemoteSigned -Scope CurrentUser

# Or, allow for the local machine (requires Administrator)

# Set-ExecutionPolicy RemoteSigned -Scope LocalMachine

Enter 'Y' to confirm if prompted.

## Setup

1. Save the network\_test.py and network\_test.ps1 files to your desired directory.

2. Install the required Python libraries if using the Python script (see Requirements).

3. Adjust PowerShell Execution Policy if necessary (see Requirements).

4. (Optional) Create your input CSV file (see format below).

## Input CSV File Format (`targets.csv`)

If using the CSV input method, create a file (e.g., targets.csv) with the following format:

* **Header Row:** Must be hostname,services
* **Data Rows:**
* Column 1: The hostname or IP address of the target.
* Column 2: A comma-separated string listing the services to test for that host (e.g., ping, http, https). Do not include spaces within the service names themselves.

**Example `targets.csv`:**

hostname,services

google.com,"ping,https"

github.com,https

192.168.1.1,ping

#internalserver.local,http # Lines starting with # are ignored

webserver.example.com,"http,https"

nonexistent.domain,ping

emptyservices.com, # This host will be skipped (no services listed)

## Usage

Run the scripts from your terminal (Command Prompt, PowerShell, Bash, etc.).

### Python (`network\_test.py`)

# --- Test a Single Host ---

python network\_test.py --host google.com --services "ping,https"

python network\_test.py --host 192.168.1.1 --services ping

# --- Test Multiple Hosts from CSV ---

python network\_test.py --csv targets.csv

# --- Test from CSV and Export Results ---

python network\_test.py --csv targets.csv --output-file network\_results.csv

python network\_test.py --host my-server --services http --outfile "C: emp\single\_test.csv" # Using alias --outfile

**Arguments:**

* --host HOST: Specify a single hostname or IP address.
* --services "service1,service2": Comma-separated services (ping, http, https) for the single host. **Required** if --host is used.
* --csv FILEPATH: Path to the input CSV file.
* --output-file FILEPATH or --outfile FILEPATH: (Optional) Path to export results to a CSV file.
* *Note: You must provide either (`--host` AND `--services`) OR `--csv`.*

### PowerShell (`network\_test.ps1`)

* \*Note: You may need to use `.

etwork\_test.ps1` if running from the current directory.\*

# --- Test a Single Host ---

.

etwork\_test.ps1 -Host google.com -Services "ping,https"

.

etwork\_test.ps1 -Host 192.168.1.1 -Services ping

# --- Test Multiple Hosts from CSV ---

.

etwork\_test.ps1 -Csv . argets.csv

# --- Test from CSV and Export Results ---

.

etwork\_test.ps1 -Csv . argets.csv -OutputFile .

etwork\_results.csv

.

etwork\_test.ps1 -Host my-server -Services http -OutputFile "C: emp\single\_test.csv"

**Parameters:**

* -Host <string>: Specify a single hostname or IP address.
* -Services <string>: Comma-separated services (ping, http, https) for the single host. **Required** if -Host is used.
* -Csv <string>: Path to the input CSV file.
* -OutputFile <string>: (Optional) Path to export results to a CSV file.
* *Note: You must provide either (`-Host` AND `-Services`) OR `-Csv`.*

## Output

### Console Output

The scripts print status updates directly to the console:

* **SUCCESS** messages are displayed in **Green**.
* **FAILED** messages are displayed in **Red**, often with details about the failure (e.g., Timeout, DNS Error, HTTP Status Code).
* **SKIPPED** or **WARNING** messages are displayed in **Yellow**.
* Target hostnames are often highlighted (e.g., Cyan).
* A summary status (per-target and overall) is printed at the end.

### CSV Output File

If the -OutputFile (PowerShell) or --output-file (Python) option is used, a CSV file will be created with the following columns:

* Timestamp: Date and time when the specific test was performed (e.g., 2025-04-05 22:43:00).
* TargetHost: The hostname or IP address that was tested.
* Service: The service that was tested (ping, http, https, or skipped).
* Status: The result of the test (SUCCESS, FAILED, SKIPPED).
* Details: Additional information about the result (e.g., HTTP Status 200, Timeout, DNS Resolution Error, Responded to ICMP echo request).

## Troubleshooting

* **Colors Not Showing:**
* Ensure you are running the script in a terminal that supports ANSI colors (e.g., Windows Terminal, PowerShell 7+, VS Code Terminal, most Linux/macOS terminals). Python's IDLE does *not* support colors.
* Make sure output is not being redirected (> or |) if you expect colors on screen.
* (Python) Ensure colorama is installed.
* **Permission Denied:**
* You might need appropriate permissions to run scripts (PowerShell Execution Policy).
* You need write permissions for the directory specified in -OutputFile/--output-file.
* **(Python) `ModuleNotFoundError: No module named 'requests'` (or `colorama`):** Run pip install requests colorama.
* **(PowerShell) Script Cannot Be Loaded / Execution Policy:** See Requirements section about Set-ExecutionPolicy.
* **Tests Failing:**
* Check for **firewalls** blocking ICMP (ping) or TCP ports 80/443 between your machine and the target.
* Verify **DNS** resolution for the hostnames.
* Ensure the target **service** (e.g., web server) is actually running on the destination host.