The Greater Wall User Manual

Intro

The Greater Wall (TGW) is a customizable powershell hunt and data collection framework. The idea behind TGW is that it can serve as a platform for scripts to be run against specified endpoints..

Language

TGW is written 100% in PowerShell. The reason for this is because in certain instances customers will either disallow the introduction of 3rd party executables or require a massive amount of justification and documentation for external tools. TGW is quite simply “just a PowerShell Script”. PowerShell is already installed on all windows systems and in most cases the administrators already use it to perform their daily tasks. Ideally, the network and the systems shouldn’t require any modifications to enable TGW to run. TGW has zero external dependencies and will work on any Windows 10 workstation.

Agentless

TGW doesn’t require an agent to be deployed to any of the workstations on the network. This simplifies the usage of TGW and doesn’t require that you install or uninstall anything before you begin, or after you’ve finished using TGW. It accomplishes its tasks through the usage of WINRM.

Framework

TGW is designed to be a framework. To add or remove functionality to TGW, no modification of the source code is necessary. Any type of code is very fragile and the smallest modification, such as the removal of a single “}” can, and in most cases will, completely break the code.

Setup and Install

To setup TGW, simply unzip the folder that was downloaded, Right-click setup.ps1 and choose “Run with PowerShell”. The Setup process should finish in about 30 seconds. Once it’s complete, an administrative PowerShell ISE window will open automatically. All files will be moved to $env:userprofile\Desktop\TheGreaterWall\.

The original .zip folder will remain in the location where you saved it during the download. If at any point you feel the need to completely start over because you’ve broken the code, or for some reason TGW isn’t working anymore, simply delete the folder on your desktop and run Setup.ps1 again from your downloads folder.

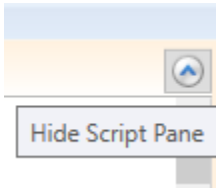
If you have results you don’t want to lose, make sure you copy them to a different location before deleting the “The Greater Wall” folder that’s on the desktop.

Using The Greater Wall

Running TGW

TGW must be ran in an administrative PowerShell ISE session. Upon initial setup this window will open automatically for you. Afterwards, you'll need to open PowerShell ISE (as administrator), file->open, and browse to the "The Greater Wall" folder on your desktop, and open TheGreaterWall.ps1.

Once TGW is open in ISE, it's helpful to hide the script plane so that your view of TGW isn't obstructed by the source code. This is a small button in the upper-right corner of your ISE window. Take note that, this is the source code for TGW; it's not advised to alter this for any reason.



Once the script plan is hidden, go ahead and hit the big green play button.

Moving in and out of The Greater Wall

As you use TGW, you will need to periodically exit and re-enter the framework. An instance where this is necessary is you've run a module and wish to navigate to the results. To do this, simply "CTRL+C" to exit TGW. Once you do this, you'll be back at the native PowerShell Prompt.

To re-enter TGW, simply type TGW in the PowerShell Prompt.

The intent is to pop in and out of the framework all in the same PowerShell ISE session.It isn't advised to close the ISE window unless you're completely done using TGW.

Setting up required information

Setting targets:

The first menu will prompt you to select a method for specifying the endpoints you're going to be querying with TGW.

Once you've done this one time, you won't be prompted for this information again. If you wish to change your targets, you must issue one of the admin commands. Those will be discussed at a later point in this document.

Testing Connectivity:

The next prompt will ask if you'd like to perform a connection test to the endpoints. This connection test consists of ICMP and an uncredentialed WINRM connection test. This step is

not necessary and depending on the configuration of the network could yield spotty results. Use caution when deciding whether or not to run the connection test. All endpoints that do not report back positively will be removed from your list of targets without your consent.

Alternatively, you can opt to not run the connection test and simply let some of the modules fail on the endpoints that TGW can't reach.

Once you've done this one time, you won't be prompted for this information again. If you wish to run the connection test at a later time, you must issue one of the admin commands. Those will be discussed at a later point in this document.

Credentials:
The next prompt will ask you to provide credentials. This is one of, if not the most important steps in setting up TGW. There is no good way to verify that the credentials you've provided are valid, so TGW will accept the credentials you supply to it. In most troubleshooting cases, the culprit ends up being the credentials.

It's advised to test the credentials on at least one remote endpoint prior to using them in TGW. This can be accomplished by issuing the following command in the PowerShell command prompt:

- Enter-pssession -computername <Hostname or IP> -credential $(get-credential)

If this is successful, you're good to go. Furthermore, you've also successfully verified that all the other configuration dependencies for remote PowerShell are in order.

Once you issue the credentials to TGW, you'll be immediately asked if you wish to reset them. This gives you the opportunity to retype them if you have a feeling that you've fat-fingered something.

Once you've done this one time, you won't be prompted for this information again. If you wish to change your credentials, you must issue one of the admin commands. Those will be discussed at a later point in this document.

Navigating through The Greater Wall

TGW is menu driven and all choices must be made by choosing a number from a list and hitting enter.

The first prompt will allow you to choose between

- Host Collection
- Event Log Collection
- Post-Process and Analyze Data

Once you've chosen an option it will take you to a new menu where you can choose the modules you'd like to run against the targets.

Module Menu

The module menu allows you to choose a module to run against the targets. Also, from this menu you'll be able to issue "admin-commands".

At the very top of this screen, you will see an overview of your settings.

Example:

Additional administrative commands are available utilizing the syntax [Admin-Commands]
[Mode: Host Collection | Credentials: Yes | Targeted Endpoints: 1]

If this overview of your settings doesn't appear to be correct, you'll need to use the appropriate admin-command to re-enter the information.

Admin Commands

Status- The status command will allow you to see the status of the modules that are being run against the targets. While the modules are being run and even after they've completed the results are only resident in memory. Nothing is written to disk until you issue the sync command. All Modules that produce errors will generate an entry in $env:userprofile\Desktop\TheGreaterWall\TGWLogs\error.log
The two primary statuses are "running" and "completed". Depending on the computational load induced by the module, some will finish quickly and some may take a while to finish.

Sync- The Sync command is crucial. If the user doesn't use the sync command, the results of the modules will not be written to disk. By issuing the sync command, it will determine which modules are complete and write those to their respective files while allowing the other modules to finish. This command will be used often and is not an automated process.

Whatis- The whatis command is essentially a man-page function. It will provide you with detailed information about each module, the commands that it uses, example outputs, etc. this command automatically places leading and trailing wildcards onto the queries. To give an example, the following two queries will work the same.

Example: whatis 4688
Example: whatis curitylog4688

In addition to that, if there are post-processed PowerShell Logs, it will allow you to view a single log entry as specified by "message hash". In the next example, if the message hash of a powershell log is 68568 and you wish to see the actual content of the powershell log you can issue the whatis command in the example below.

Example: whatis 68568

Show-creds- This will show you the username you have specified and whether or not credentials have been supplied. It will not show you plain-text credentials.

Reset-creds- Allows you to reset the provided credentials

Show-targets- Shows a list of all users specified target IPs or Hostnames

Add-target- Allows a user to add a target to the already existing list of targets

Remove-target- Allows a user to remove a target from the already existing list of targets

Reset-targets- Resets the target list and prompts for new targets

Hail-mary- This will allow the user to run all modules on all targets. It's advised to use caution with this. Depending on how many modules are available and how many targets you have you may run into issues with system resources and over-tax your available memory/CPU.

Go-interactive- This will allow you to specify or choose from a list of targets to go interactive on. Upon specifying a target, a separate PowerShell window will spawn and you'll be remoted into the target. This can be useful for troubleshooting or to accomplish simple tasks on the target that aren't offered by any of the modules.

Run-connectiontest- This will conduct a ping scan and uncredentialed winrm scan against the targets. Depending on the configuration of the network, this has potential to produce false negatives. Use caution as any endpoint that doesn't report back will be removed from the target list without prompting the user.

Show-connectionstatus- If the connection test was ran, this will show you an overview of the reachable and unreachable targets.

archive-results- If your $env:userprofile\Desktop\TheGreaterWall\Results folder is full of old results that you don't want to get mixed with what you're currently working on, you can archive-results. This will simply place them in a .zip folder, making them available for later viewing while allowing the post-processor to ignore them.

Module-status- This will show each module and whether or not it has a valid entry in modules.conf. If it's not present in modules.conf, it will still work, but it can't be post-processed.

Back- Used simply to navigate. For example, if you're at the host collection menu and wish to go to the event log collection menu, you would type "back", then choose the option for the event log collection menu.

<u>External admin-commands</u>
There are a couple of commands that can be issued from the native powershell prompt in your ISE session. These are called raw commands and you have the option to use the "-rawcommand" switch option.

Ex: tgw -rowcommand sync
Or
Ex: tgw sync

Below is a list of those commands
Status- Shows status of running modules
Status-watch- This command will generate a constantly refreshing status of the running modules. This is useful if you want to sit and watch your modules as they complete.
Sync- Receives the results of the module and writes them to their appropriate files.
Archive-results- Zips old results and saves them
Reset- This will completely reset TheGreaterWall and all specified options. This should only be used as a last ditch effort for troubleshooting, aside from re-running setup.ps1
Postprocess- Runs the post processor.

Post-Processor

The post-processing functionality of The Greater Wall is quite simply, outlier analysis. It will determine which things show up on less than half of the endpoints. The results of the outlier analysis be in
$env:userprofile\Desktop\TheGreaterWall\Results\Post-processed*\AnalysisResults\OutlierAnalysis

Uponi running the post processor, a new folder will be created. In this folder you will have enriched data for each target, raw data, and outlier analysis.

The enriched data for each host is pretty much the same info as the original data except there the IP field has been populated with the IP or Hostname of the target.

The raw data is a complete concatenated .csv of all data from every host, on .csv per dataset.

Examples:
all_smb.csv
All_processinfo.csv
etc.

This raw data can be extremely useful and is easy to manipulate in MS Exec with pivot tables.

The Outlier analysis data is entries from the raw data that were observed to only be present on less than half of the queried endpoints. This number is automatically calculated, therefore if you have data from 10 endpoints, the post processor will know to look for things that showed up only on 5 or less endpoints.

If you've created a custom module and adhered to the guidelines of how to write it, and wish to have it post-processed, you must create an entry in modules.conf for that module.

Modules.conf will be the way that the post processor knows the headers for the information and what exactly to look for when determining outliers. This is important because there are only certain things about a dataset that can and should be analyzed for outliers, an example of something you wouldn't want to analyze for outliers would be a timestamp, or a UID.

This is an example entry from modules.conf. As you can see, there are 3 main components: Pivot, IP, and csvheader.

Pivot is the field that you wish to analyze for outliers, put as many as you'd like but ensure that whatever you put in here, matches what is present in your data. If in your custom module, you named one of the fields "taskname", ensure that's exactly what you put in modules.conf, don't put "nameoftask" or anything else that isn't an exact match.

IP is specifying what you named the field in your module that is indicative of the endpoint.

Csvheader is derived directly from your module. Whatever your CSV header is, paste it in here.

tasksscheduled:pivot:taskname
tasksscheduled:pivot:taskpath
tasksscheduled:pivot:action
tasksscheduled:IP:IP
tasksscheduled:csvheader:IP,Hostname,DateCollected,Task Name,Task State,Task Path,Action,Author

Troubleshooting and Configuration dependencies

WinRM

Winrm must be enabled on all endpoints In order for TGW to communicate to/from them. Winrm will use Port(s) 5985 and 5986.

This can be checked by issuing the following command:

- Get-service -name Winrm

If the service is stopped, you can start it by issuing the following commands

- Start-Service -name Winrm
- Enable-psremoting -force

The above method is something that must be done on the computer that you're running TGW from as well as all the endpoints that you wish to remotely query. If WinRM is not enabled on the distant endpoints, you won't be able to enable it remotely. You'll need to contact an administrator who has the ability to modify the domain and have them enable WinRM on the endpoints. If you have physical access to the endpoints, and administrative credentials, you can manually start the service.

Trusted Hosts

In some instances, the TrustedHosts value may need to be modified in order to allow the remote PowerShell connection to be successful.

This can be check by issuing the following command:

- get-item wsman:\localhost\client\trustedhosts

If the value is blank, it can set by issuing the following command:

- Set-item wsman:\localhost\client\trustedhosts –value *
- Note: Setting this value to "*" isn't recommended but in a pinch as a temporary means of making things work, it can be useful. The recommended method would be to set this value to the IP address of the workstation you're running TGW from.

Firewall Profile

Depending on the network you're on, and especially if you're testing TGW on your LAN you may need to modify the firewall profile to allow remote PowerShell.

This can be accomplished by issuing the following command:

- Set-netConnectionprofile –networkcategory private

Language mode

There are a variety of factors that will cause TGW to fail to run, or not run properly. One of these is Language mode. The language mode for PowerShell must be set to "FullLanguageMode".

This can be checked by issuing the following command:

- $ExecutionContext.SessionState.LanguageMode

The language mode can be adjusted by issuing the following command

- $ExecutionContext.SessionState.LanguageMode = 'fulllanguage'

For some configuration changes, you may still get "Access Denied" errors even though you're using administrative credentials. If this is the case, try altering local group policy for the setting that you're trying to change. If that can't be done, you'll probably need to contact an administrator who has the ability to change the setting in Active Directory.

Execution Policy

The PowerShell Execution Policy must be set to unrestricted.

This can be checked by issuing the following command:

- Get-ExecutionPolicy

The Execution Polocy can be adjusted by issuing the following command

- Set-ExecutionPolicy -scope currentuser unrestricted

For some configuration changes, you may still get “Access Denied” errors even though you’re using administrative credentials. If this is the case, try altering local group policy for the setting that you’re trying to change. If that can’t be done, you’ll probably need to contact an administrator who has the ability to change the setting in Active Directory.

Adding a module

All modules are .psm1 files. If you wish to add a module, you must first write the script and ensure that it follows all necessary rules for a PowerShell Module. The method that this has been done for all current modules is as follows.

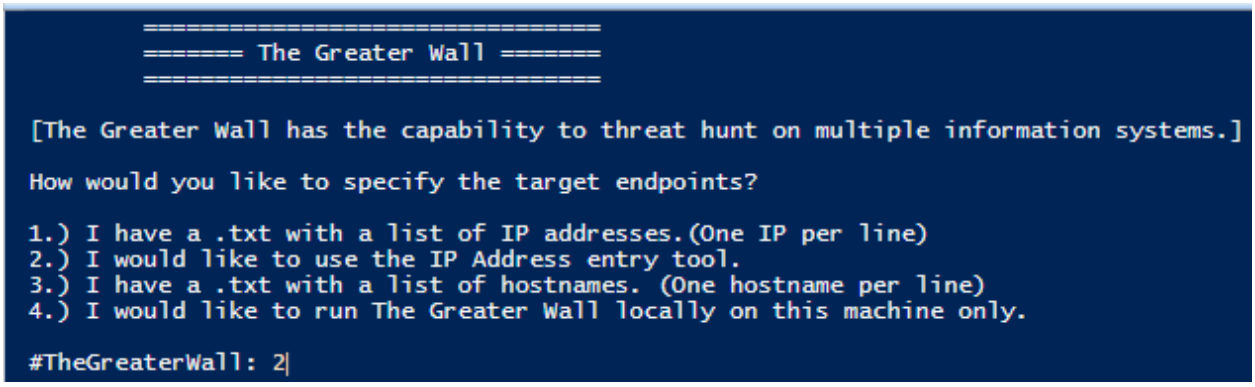
1. Write the script and ensure that it’s wrapped in a function
2. Ensure that at the end of the script, it outputs to STDOUT in CSV format
3. Ensure to include “export-modulemember -function <Name of function>” at the very bottom of the script.
4. Save the .psm1 file with the same name as the function. For example if your function is named “CollectAllData”, name your file “CollectAllData.psm1”.
5. Depending on whether you consider the module to be “Host Collection” or “Event Log Collection”, browse to that folder (ex: $env:userprofile\Desktop\TheGreaterWall\Modules\HostCollection”)
6. Create a new folder and name it the same name as your function. If your function is called “CollectAllData”, name the folder “CollectAllData”.
7. Place your .psm1 file into that folder and make sure it’s named correctly. If your function is called “CollectAllData”, name the file “CollectAllData.psm1”.
8. That’s it. TGW will automatically recognize your new module and it will become available in the menu immediately. If you’d like your module to be post-processed and analyzed for outliers, there’s a few more things to do. That will be discussed in the Post-Processor section of this manual.

The easiest and quickest way to make new modules is to use an existing one as a template, modify it, and save it under a different filename to avoid overwriting the original module. See example below of a simple Module.

```
function smb{
   function build-class{
      $outputclass= [pscustomobject][ordered]@{
         IP= "null"
         Hostname= $null
```

```
        DateCollected= $null
        SMBv1= $null
        SMBv2= $Null
      }
    return $outputclass
    }

    $output= @()
    $results= build-class

    $hostname= $env:COMPUTERNAME
            $SMBVersion1 = (Get-SmbServerConfiguration).EnableSMB1Protocol
            $SMBVersion2= (Get-SmbServerConfiguration).EnableSMB2Protocol
    $date= (Get-Date -Format "dd-MMM-yyyy HH:mm").Split(":") -join ""

    $results.Hostname= $hostname
    $results.DateCollected= $date
    $results.SMBv1= $SMBVersion1
    $results.SMBv2= $SMBVersion2

    $output+= $results | ConvertTo-Json
            $output | ConvertFrom-Json | ConvertTo-Csv -NoTypeInformation
            }

export-modulemember -function smb
```

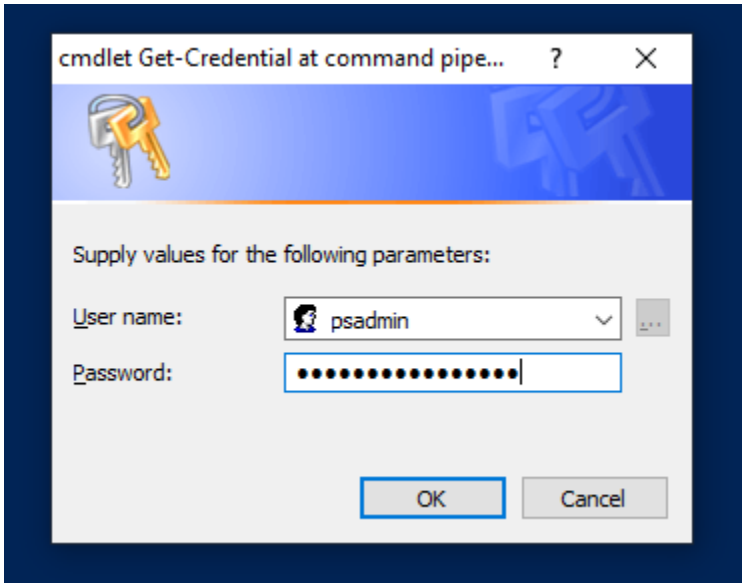
Usage Examples / walkthrough



```
        ================================
        ======= The Greater Wall =======
        ================================

Please provide the IP Address(es) you would like to investigate:

    Example: x.x.x.x, x.x.x.x, x.x.x.x

    Example: x.x.x.x/24

    Example: x.x.x.[1-15]

    Example: x.x.x.x,x.x.x.x/24,x.x.x.[1-15]

#TheGreaterWall: 10.50.66.2/26
```

```
        ================================
        ======= The Greater Wall =======
        ================================

           *****Connection Test*****

1.) Test the connectivity of your (62) IP(s).
2.) Test the connectivity of a Single IP.
3.) Do not run Connection test.
 : 3
```

```
        ================================
        ======= The Greater Wall =======
        ================================

[Warning] No Credentials detected
Do you wish to use credentials?

1.) Yes
2.) No

#TheGreaterWall: 1
```



```
        ================================
        ======= The Greater Wall =======
        ================================

Credentials saved. Do you want to reset them?
1.) Yes
2.) No
#TheGreaterWall: 2
```

```
        ================================
        ======= The Greater Wall =======
        ================================

Choose what type of actions you wish to perform

1.) Host Collection
2.) EventLog Collection
3.) Post-process and analyze data
 : 1
```

```
        ================================
        ======= The Greater Wall =======
        ================================

Additional administrative commands are available utilizing the syntax [Admin-Commands].

[ Mode: Host Collection | Credentials: Yes | Targeted Endpoints: 62 ]

Select a module to run against targeted endpoints:

    1.) ActiveDirectoryEnumeration
    2.) AlternateDataStreams
    3.) CrashedApplications
    4.) DllInformation
    5.) EnumerateUSB
    6.) HotFixes
    7.) ImageFileExecutionOptions
    8.) InstalledSoftware
    9.) NetworkConnections
    10.) persistence
    11.) Prefetch
    12.) ProcessInfo
    13.) ProcessTree
    14.) ServiceInfo
    15.) smb
    16.) TasksScheduled

#TheGreaterWall: 16
```

```
        ================================
        ======= The Greater Wall =======
        ================================

Additional administrative commands are available utilizing the syntax [Admin-Commands].

[ Mode: Host Collection | Credentials: Yes | Targeted Endpoints: 62 ]

Select a module to run against targeted endpoints:

    1.) ActiveDirectoryEnumeration
    2.) AlternateDataStreams
    3.) CrashedApplications
    4.) DllInformation
    5.) EnumerateUSB
    6.) HotFixes
    7.) ImageFileExecutionOptions
    8.) InstalledSoftware
    9.) NetworkConnections
    10.) persistence
    11.) Prefetch
    12.) ProcessInfo
    13.) ProcessTree
    14.) ServiceInfo
    15.) smb
    16.) TasksScheduled

#TheGreaterWall: status
```

```
--STATUS OF RUNNING TASKS--


Name                                          State   HasData
----                                          -----   -------
10.50.66.1-TasksScheduled-10-Nov-2021 0855  Running Undetermined
10.50.66.2-TasksScheduled-10-Nov-2021 0855  Running Undetermined
10.50.66.3-TasksScheduled-10-Nov-2021 0855  Running Undetermined
10.50.66.4-TasksScheduled-10-Nov-2021 0855  Running Undetermined
10.50.66.5-TasksScheduled-10-Nov-2021 0855  Running Undetermined
10.50.66.6-TasksScheduled-10-Nov-2021 0855  Running Undetermined
10.50.66.7-TasksScheduled-10-Nov-2021 0855  Running Undetermined
10.50.66.8-TasksScheduled-10-Nov-2021 0855  Running Undetermined
10.50.66.9-TasksScheduled-10-Nov-2021 0855  Running Undetermined
10.50.66.10-TasksScheduled-10-Nov-2021 0855 Running Undetermined
```

```
Select a module to run against targeted endpoints:

    1.) ActiveDirectoryEnumeration
    2.) AlternateDataStreams
    3.) CrashedApplications
    4.) DllInformation
    5.) EnumerateUSB
    6.) HotFixes
    7.) ImageFileExecutionOptions
    8.) InstalledSoftware
    9.) NetworkConnections
    10.) persistence
    11.) Prefetch
    12.) ProcessInfo
    13.) ProcessTree
    14.) ServiceInfo
    15.) smb
    16.) TasksScheduled

#TheGreaterWall: sync
```

CTRL+C

```
PS C:\Users\user1\desktop\thegreaterwall> cd results
```

```
PS C:\Users\user1\desktop\thegreaterwall\results> Get-ChildItem
```

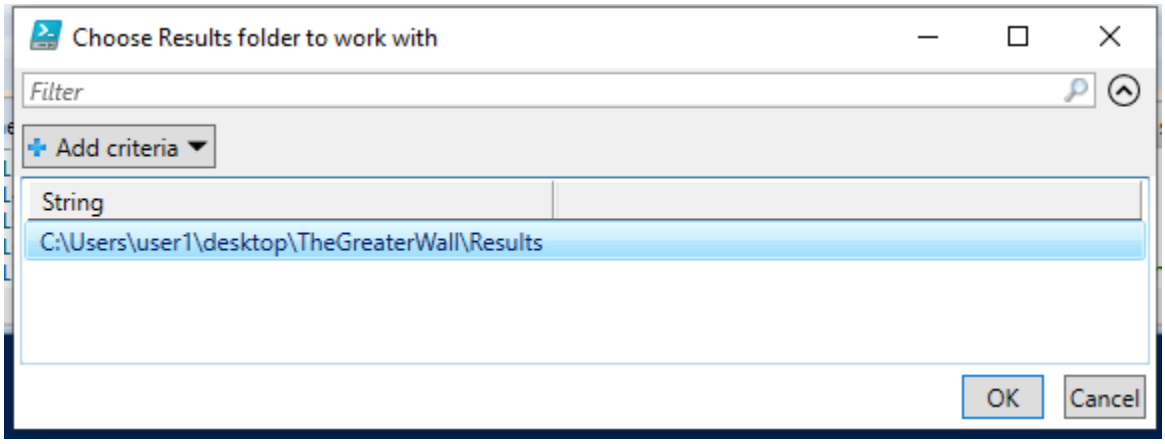
```
PS C:\Users\user1\desktop\thegreaterwall\results> Get-ChildItem


    Directory: C:\Users\user1\desktop\thegreaterwall\results


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        11/10/2021   9:01 AM                10.50.66.1
d-----        11/10/2021   9:01 AM                10.50.66.2
d-----        11/10/2021   9:01 AM                10.50.66.3
d-----        11/10/2021   8:59 AM                Archived_results
```

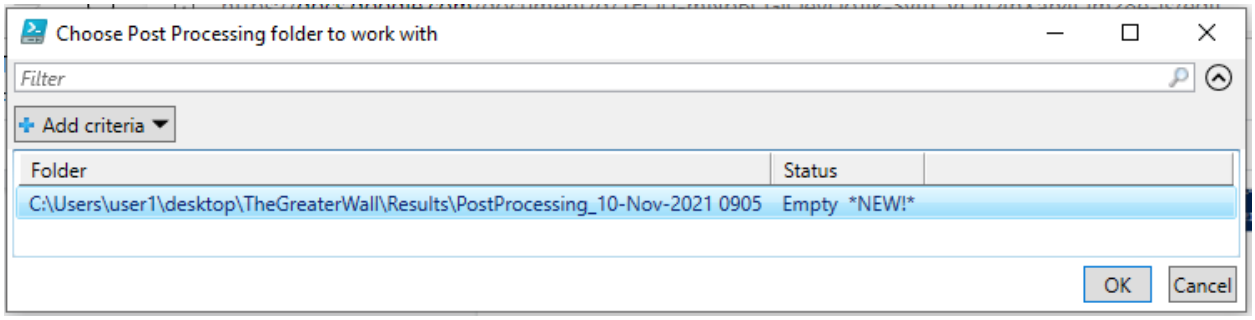
```
PS C:\Users\user1\desktop\thegreaterwall\results> cd .\10.50.66.1
```

```
PS C:\Users\user1\desktop\thegreaterwall\results\10.50.66.1> Get-ChildItem


    Directory: C:\Users\user1\desktop\thegreaterwall\results\10.50.66.1


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        10/15/2021   2:24 PM         119362 127.0.0.1-AlternateDataStreams-15-Oct-2021 1402.txt
-a----        10/15/2021   2:24 PM          57130 127.0.0.1-CrashedApplications-15-Oct-2021 1402.txt
-a----        10/15/2021   2:24 PM        6678766 127.0.0.1-DllInformation-15-Oct-2021 1402.txt
-a----        10/15/2021   2:24 PM          40592 127.0.0.1-EnumerateUSB-15-Oct-2021 1402.txt
-a----        10/15/2021   2:24 PM           1976 127.0.0.1-HotFixes-15-Oct-2021 1402.txt
-a----        10/15/2021   2:24 PM           4582 127.0.0.1-InstalledSoftware-15-Oct-2021 1402.txt
-a----        10/15/2021   2:24 PM          17658 127.0.0.1-NetworkConnections-15-Oct-2021 1402.txt
-a----        10/15/2021   2:24 PM           4166 127.0.0.1-persistence-15-Oct-2021 1402.txt
-a----        10/15/2021   2:24 PM          47968 127.0.0.1-Prefetch-15-Oct-2021 1402.txt
-a----        10/15/2021   2:24 PM         193542 127.0.0.1-ProcessInfo-15-Oct-2021 1402.txt
-a----        10/15/2021   2:24 PM          16170 127.0.0.1-ProcessTree-15-Oct-2021 1402.txt
-a----        10/15/2021   2:24 PM          92772 127.0.0.1-ServiceInfo-15-Oct-2021 1402.txt
-a----        10/15/2021   2:24 PM            230 127.0.0.1-smb-15-Oct-2021 1402.txt
-a----        10/15/2021   2:24 PM          66272 127.0.0.1-TasksScheduled-15-Oct-2021 1402.txt
```

```
PS C:\Users\user1\desktop\thegreaterwall\results\10.50.66.1> tgw
```

```
        ================================
        ======= The Greater Wall =======
        ================================

Choose what type of actions you wish to perform

1.) Host Collection
2.) EventLog Collection
3.) Post-process and analyze data
 : 3
```






```
PS C:\Users\user1\desktop\thegreaterwall\results> Get-ChildItem


d-----        11/10/2021   9:05 AM                PostProcessing_10-Nov-2021 0905


PS C:\Users\user1\desktop\thegreaterwall\results> cd '.\PostProcessing_10-Nov-2021 0905'
PS C:\Users\user1\desktop\thegreaterwall\results\PostProcessing_10-Nov-2021 0905> Get-ChildItem


d-----        11/10/2021   9:05 AM                AnalysisResults
d-----        11/10/2021   9:05 AM                Archived_results-PostProcessed
d-----        11/10/2021   9:05 AM                RawData


PS C:\Users\user1\desktop\thegreaterwall\results\PostProcessing_10-Nov-2021 0905> cd .\RawData
PS C:\Users\user1\desktop\thegreaterwall\results\PostProcessing_10-Nov-2021 0905\RawData> Get-ChildItem


    Directory: C:\Users\user1\desktop\thegreaterwall\results\PostProcessing_10-Nov-2021 0905\RawData


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        11/10/2021   9:05 AM         596342 all_alternatedatastreams.csv
-a----        11/10/2021   9:05 AM         284798 all_CrashedApplications.csv
-a----        11/10/2021   9:05 AM       33393070 all_dllinformation.csv
-a----        11/10/2021   9:05 AM         201588 all_EnumerateUSB.csv
-a----        11/10/2021   9:05 AM           9436 all_hotfixes.csv
-a----        11/10/2021   9:05 AM          22204 all_InstalledSoftware.csv
-a----        11/10/2021   9:05 AM          20354 all_persistence.csv
-a----        11/10/2021   9:05 AM         239372 all_Prefetch.csv
-a----        11/10/2021   9:05 AM         965474 all_processinfo.csv
-a----        11/10/2021   9:05 AM         462672 all_serviceinfo.csv
-a----        11/10/2021   9:05 AM            730 all_smb.csv
-a----        11/10/2021   9:05 AM         330646 all_tasksscheduled.csv


PS C:\Users\user1\desktop\thegreaterwall\results\PostProcessing_10-Nov-2021 0905> cd .\AnalysisResults
PS C:\Users\user1\desktop\thegreaterwall\results\PostProcessing_10-Nov-2021 0905\AnalysisResults> Get-ChildItem


    Directory: C:\Users\user1\desktop\thegreaterwall\results\PostProcessing_10-Nov-2021 0905\AnalysisResults


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        11/10/2021   9:05 AM                OutlyerAnalysis


PS C:\Users\user1\desktop\TheGreaterWall\Results\PostProcessing_10-Nov-2021 0917\AnalysisResults> cd .\OutlyerAnalysis
gc .\tasksscheduled-Analysis.csv | convertfrom-csv
```

```
IP              : 127.0.0.2
Hostname        : LAPTOP-D4J1KGC0
DateCollected   : 15-Oct-2021 1402
Task Name       : Adobe Acrobat Update Taskmalware
Task State      : 3
Task Path       : \
Action          : C:\Program Files (x86)\Common Files\Adobe\ARM\1.0\AdobeARMalware.exe
Author          : Adobe Systems Incorporated
PropertyFlagged : action
```