

Powershell remote plugins

The GESA_PWSH_REMOTE and CHECK_PWSH_REMOTE plugins enable Windows scripts/executables to be run remotely and their screen output (stdout) and the value returned by them to be retrieved. These remote scripts can be signed for added security.

This is how the Esia plugin works: it asks for the following parameters:

- The name or path of the script to be executed
- A string containing a series of parameters to be passed to your powershell script.

You need to have set the authentication parameters in esia and activate powershell on the remote machine: [Activating powershell in Esia](#)

With CHECK_PWSH_REMOTE

Here are the default settings:

```
-H $IP -A $AUTH_FILE_PASSWORD -T remote -p <path> -a "<args>"
```

And an example of settings :

```
-H $IP -A $AUTH_FILE_PASSWORD -T remote -p "c:\plugins\mypluginvalue.ps1" -a "-warning 10 -critical 70"
```

Or :

```
-H $IP -A $AUTH_FILE_PASSWORD -T remote -p "c:\plugins\mypluginvalue.ps1"
```

The password must be configured via the device's advanced configuration settings.

With GESA_PWSH_REMOTE

Here are the default settings:

```
-u -S $GESASN -t 50 -c check_pwsh_remote -a main.pwsh $IP <path> "<args>"
```

And an example of settings:

```
-u -S $GESASN -t 50 -c check_pwsh_remote -a main.pwsh $IP  
"c:\plugins\mypluginvalue.ps1"
```

Or :

```
-u -S $GESASN -t 50 -c check_pwsh_remote -a main.pwsh $IP
```

```
"c:\plugins\mypluginvalue.ps1" '"-warning 10 -critical 70"'
```

Or else :

```
-u -S $GESASN -t 50 -c check_pwsh_remote -a main.pwsh $IP  
"c:\plugins\mypluginvalue.ps1" '"10 70"'
```

The password must be configured on the Unity

Explanation of parameters

- \$IP: The IP address of the machine containing the script/runtime to be executed.
- <path> The path to this executable
- <args> : The arguments to provide. **Be careful to put (quote and double quote in the right order), see examples above and below.** They are necessary.

```
'"<args>"'
```

Writing a powershell script that can be used by the plugin

See here for information on how Esia plugins work: [How to create a plugin](#)

The powershell plugins used by GESA_PWSH_REMOTE and CHECK_PWSH_REMOTE work in the same way.

The examples below are in powershell, but you can use any other language.

Plugin that always returns OK :

Settings :

```
'c:\plugins\alreadyOK.ps1'
```

Or

```
'c:\plugins\alreadyOK.ps1' ""
```

Script content :

```
Write-Output "Hello world"  
exit 0;
```

Compares a value and returns WARNING or CRITICAL :

Parameterization :

```
'c:\plugins\mypluginvalue.ps1' '"-warning 10 -critical 70"'
```

Or :

```
'c:\plugins\mypluginvalue.ps1' '"10 70"'
```

Script content : <ignore>

```
Param(
    [Parameter(Mandatory=$True)]
    [String] $warning,
    [Parameter(Mandatory=$True)]
    [String] $critical
)

$value = 50;

if( $value -gt $critical )
{
    Write-Output "CRITICAL : value is $value (> $critical)"
    exit 2;
}

if( $value -gt $warning )
{
    Write-Output "WARNING : value is $value (> $warning)"
    exit 1;
}
Write-Output "OK : value is $value (<ignore> < $warning)"
exit 0;
```

</ignore></ignore>

From:

<https://wiki.esia-sa.com/> - **Esia Wiki**

Permanent link:

https://wiki.esia-sa.com/en/protocols/powershell_remote_plugin

Last update: **2025/05/12 12:48**

