



# INTEGRATING SYHUNT WITH POWERSHELL

The information in this document applies to **version 6.8.6** of Syhunt Hybrid.

Syhunt scans can be easily executed from within a PowerShell script, allowing you to integrate the Syhunt Dynamic, Syhunt Code e Syhunt Mobile tools into your continuous delivery pipeline, CLI and more. The Syhunt Hybrid setup automatically installs a Syhunt PowerShell module in Windows (10 / 7) with functions that enable you to start scans, get reports and results and perform pass/fail testing.

```
Windows PowerShell
PS C:\DevTemp\test>
PS C:\DevTemp\test> Start-CodeScan -output "report.pdf"
Preparing to scan target: C:\DevTemp\test
SYHUNT CODESCAN 6.8.6 PLATINUM PLUS 1-YEAR (c) 2020 Syhunt

VULNERABLE!
Found 2 vulnerabilities
The following scripts are vulnerable:
case008-xss.php

Generating report...
Saved to C:\DevTemp\test\report.pdf.
```

## ADDING SYHUNT TO YOUR POWERSHELL SCRIPT

Insert the below code at the appropriate position of your PowerShell script:

```
# SAST Example - Scan local directory/repository
Start-CodeScan -pfcond "fail-if:risk=mediumup"

# SAST Example - Scan a remote GIT repository
$MyProject = @{
    target = 'https://github.com/syhunt/vulnphp.git';
    branch = 'master';
    pfcond = 'fail-if:risk=mediumup';
    output = 'report.pdf'
}
Start-CodeScan @MyProject

# DAST Example - Scan URL
$MyWebsite= @{
    target = 'https://www.somewebsite.com';
    pfcond = 'fail-if:risk=mediumup';
    output = 'report.pdf'
}
Start-DynamicScan @MyWebsite
```

## START-DYNAMICSCAN FUNCTION

Syhunt Dynamic must be launched through the Start-DynamicScan() function. The following parameters must be provided when calling the Start-DynamicScan() function:

- **target** (required) - the target URL to be scanned (eg. <http://www.somesite.com>)
- **huntmethod** (optional) - the **Hunt Method** to be used during the scan, If omitted, the default method will be used.
- **pfcond** (optional) - allows the script to fail with proper exit code if a certain condition is met. See below a list of available pass/fail conditions.
- **output** (optional) - allows to set an output filename (eg. report.pdf or report.html).
- **outputex** (optional) - allows to set a second output filename (eg. export.json).
- **verbmode** (optional) - \$false by default. If changed to true, turns on verbose mode allowing information other than error and basic info to be printed.
- **genrep** (optional) - \$true by default. If changed to false, Syhunt will not generate an output file.
- **redirIO** (optional) - \$true by default. If changed to false, input and output from the Syhunt scanner will not be redirected to the console.

When using the output or outputex parameters, all output formats supported by Syhunt are available. The report or export will be saved to the current working directory, unless a full path name is provided.

Examples:

```
# Example 1 - Scan URL with single line
Start-DynamicScan -target 'https://www.somewebsite.com' -pfcond 'fail-if:risk=mediumup'

# Example 2 - Scan URL
$MyWebsite= @{
    target = 'https://www.somewebsite.com';
    pfcond = 'failifriskmedium';
    output = 'report.pdf'
}
Start-DynamicScan @MyWebsite
```

## START-CODESCAN FUNCTION

Syhunt Code must be launched through the Start-CodeScan() function. The following parameters can be provided when calling the Start-CodeScan() function, all of which are optional:

- **target** - the target URL of a GIT repository to be scanned, or a local source code directory or file. If the target parameter is omitted, the current working directory is scanned.

- **branch** - the repository branch to be scanned. If the branch parameter is omitted, the master branch is scanned.
- **huntmethod** - the **Hunt Method** to be used during the scan, If omitted, the default method will be used.
- **pfcond** - allows the script to fail with proper exit code if a certain condition is met. See below a list of available pass/fail conditions.
- **output** - allows to set an output filename (eg. report.pdf or report.html).
- **outputex** - allows to set a second output filename (eg. export.json).
- **verbmode** - \$false by default. If changed to true, turns on verbose mode allowing information other than error and basic info to be printed.
- **genrep** - \$true by default. If changed to false, Syhunt will not generate an output file.
- **redirIO** - \$true by default. If changed to false input and output from the Syhunt scanner will not be redirected to the console.

When using the output or outputex parameters, all output formats supported by Syhunt are available. The report or export will be saved to the current working directory, unless a full path name is provided.

Examples:

```
# Example 1 - Scan the current directory/repository
Start-CodeScan -pfcond "fail-if:risk=mediumup"

# Example 2 - Scan a remote GIT project
Start-CodeScan -target "https://github.com/someuser/somerepo.git" -huntmethod "normal" -pfcond "fail-if:risk=mediumup"

# Example 3 - Scan a specific local directory
Start-CodeScan -target "C:\www\" -huntmethod "normal" -pfcond "fail-if:risk=mediumup"
```

## PASS/FAIL CONDITIONS

The following are the pass/fail conditions currently supported by Syhunt:

- `fail-if:risk=high` - Fail if a High risk vulnerability is found
- `fail-if:risk=mediumup` - Fail if a Medium or High risk vulnerability is found
- `fail-if:risk=lowup` - Fail if a Low, Medium or High risk vulnerability is found

## ADDITIONAL NOTES

If you get the error **Script.ps1 cannot be loaded because running scripts is disabled on this system**, launch PowerShell with administrative rights and enter the following command to enable scripting:

```
set-executionpolicy remotesigned
```

If you get the error **The term 'Start-CodeorDynamicScan' is not script file, or operable program.**, you need to load Syhunt from within PowerShell. Just call `Import-Module Syhunt` and call the scan function again.

## AUTO-LOADING SYHUNT

In Windows 10, Syhunt should be automatically loaded.

In Windows 7, you can manually load Syhunt every time by calling **Import-Module Syhunt** before calling Syhunt functions or you can auto-load Syhunt - add Syhunt to your PowerShell profile and Syhunt will be loaded automatically:

1. Go to the PowerShell profile path, which by default is: `C:\Documents and Settings\User\My Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1`
2. Create **Microsoft.PowerShell\_profile.ps1** file
3. Add the line **Import-Module Syhunt** and save the file

---

For additional product documentation, visit [syhunt.com/docs](https://syhunt.com/docs)

