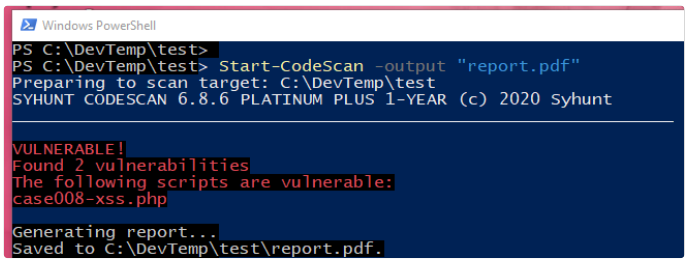
INTEGRANDO O SYHUNT AO POWERSHELL

As informações contidas neste documento se aplicam a **versão 6.9.17** do Syhunt Hybrid.

As verificações do Syhunt podem ser facilmente executadas a partir de um script do PowerShell, permitindo integrar as ferramentas Syhunt Dynamic, Syhunt Code e Syhunt Mobile ao seu processo de entrega contínua, CLI e muito mais. A instalação do Syhunt Hybrid instala automaticamente um módulo Syhunt no PowerShell do Windows (10/7) com funções que permitem iniciar varreduras, obter relatórios e resultados e executar testes de aprovação / reprovação.



ADICIONANDO O SYHUNT EM SEU SCRIPT POWERSHELL

Insira o código abaixo na posição apropriada do seu script do PowerShell:

```
# Exemplo de SAST - Analisar diretório / repositório local
Start-CodeScan -pfcond "medium"

# Exemplo de SAST - Analisar um repositório GIT remoto
$MyProject = @{
  target = 'https://github.com/syhunt/vulnphp.git';
  branch = 'main';
  pfcond = 'medium';
  output = 'report.pdf'
}
Start-CodeScan @MyProject
```

```
# Exemplo de DAST - Analisar um URL
$MyWebsite= @{
  target = 'https://www.somewebsite.com';
  pfcond = 'medium';
  output = 'report.pdf'
}
Start-DynamicScan @MyWebsite
```

START-DYNAMICSCAN - INICIANDO UMA VARREDURA DINÂMICA

O Syhunt Dynamic deve ser iniciado através da função Start-DynamicScan(). Os seguintes parâmetros devem ser fornecidos ao chamar a função Start-DynamicScan():

- **target** (obrigatório) - o URL alvo a ser analisado (ex. http://www.somesite.com)
- **huntmethod** (opcional) - o Método de Varredura a ser usado durante a análise. Se omitido, o método padrão será usado.
- **pfcond** (opcional) - permite que o script falhe com o código de saída adequado se uma determinada condição for atendida. Veja abaixo uma lista das condições de aprovação / reprovação disponíveis.
- **tracker** (opcional) - o nome do rastreador criado anteriormente ou um rastreador gerado dinamicamente para o qual será enviado um resumo das vulnerabilidades identificadas ao final da varredura. Exemplos
- **output** (opcional) - permite definir um nome de arquivo de relatório (por exemplo, report.pdf ou report.html).
- **outputex** (opcional) - permite definir um segundo nome de arquivo de saída (por exemplo, export.json).
- **verbmode** (opcional) - $ false por padrão. Se alterado para true, ativa o modo detalhado, permitindo que informações de erro e informações básicas sejam adicionadas ao console.
- **genrep** (opcional) - $ true por padrão. Se alterado para false, o Syhunt não gerará um arquivo de relatório.
- **redirIO** (opcional) - $ true por padrão. Se alterado para false, informações de status em tempo-real do scanner Syhunt não serão redirecionadas para o console.
- **timelimit** (opcional) - define o tempo máximo da varredura (padrão: sem limite). Caso o tempo seja atingido, a varredura é cancelada. Exemplos: 1d, 3h, 2h30m, 50m

Ao usar os parâmetros output ou outputex, todos os formatos de saída suportados pelo Syhunt estão disponíveis. O relatório ou exportação será salvo no diretório de trabalho atual, a menos que seja fornecido um nome de caminho completo.

Exemplos:

```
# Exemplo 1 - Analisar um URL com uma única linha
Start-DynamicScan -target 'https://www.somewebsite.com' -pfcond 'fail-if:risk=mediumup'

# Exemplo 2 - Analisar um URL
$MyWebsite= @{
  target = 'https://www.somewebsite.com';
  pfcond = 'medium';
  output = 'report.pdf'
}
Start-DynamicScan @MyWebsite
```

START-CODESCAN - INICIANDO UMA VARREDURA DE CÓDIGO-FONTE

O Syhunt Code deve ser iniciado através da função Start-CodeScan(). Os seguintes parâmetros podem ser fornecidos ao chamar a função Start-CodeScan(), sendo todas eles opcionais:

- **target** - o URL de um repositório GIT ou um diretório ou arquivo de código-fonte local a ser verificado. Se o parâmetro target for omitido, o diretório de trabalho atual será verificado.
- **branch** - a ramificação do repositório a ser analisada. Se o parâmetro branch for omitido, o branch padrão será analisado.
- **huntmethod** - o Método de Varredura a ser usado durante a análise. Se omitido, o método padrão será usado.
- **pfcond** - permite que o script falhe com o código de saída adequado se uma determinada condição for atendida. Veja abaixo uma lista das condições de aprovação / reprovação disponíveis.
- **output** - permite definir um nome de arquivo de relatório (por exemplo, report.pdf ou report.html).
- **outputex** - permite definir um segundo nome de arquivo de saída (por exemplo, export.json).
- **verbmode** - $ false por padrão. Se alterado para true, ativa o modo detalhado, permitindo que informações de erro e informações básicas sejam adicionadas ao console.
- **genrep** - $ true por padrão. Se alterado para false, o Syhunt não gerará um arquivo de relatório.
- **redirIO** - $ true por padrão. Se alterado para false, informações de status em tempo-real do scanner Syhunt não serão redirecionadas para o console.
- **timelimit** (opcional) - define o tempo máximo da varredura (padrão: sem limite). Caso o tempo seja atingido, a varredura é cancelada. Exemplos: 1d, 3h, 2h30m, 50m

Ao usar os parâmetros output ou outputex, todos os formatos de saída suportados pelo Syhunt estão disponíveis. O relatório ou exportação será salvo no diretório de trabalho atual, a menos que seja fornecido um nome de caminho completo.

Exemplos:

```
# Exemplo 1 - Analisando o diretório / repositório atual
Start-CodeScan -pfcond "medium"

# Exemplo 2 - Analisando um projeto GIT remoto
Start-CodeScan -target "https://github.com/someuser/somerepo.git" -huntmethod "normal" -pfcond "medium"

# Exemplo 3 - Analisando um diretório local específico
Start-CodeScan -target "C:\www\" -huntmethod "normal" -pfcond "medium"
```

CONDIÇÕES DE APROVAÇÃO / REPROVAÇÃO

A seguir estão as condições de aprovação / reprovação atualmente suportadas pelo Syhunt:

- `high` ou `fail-if:risk=high` - Falha se for encontrada uma vulnerabilidade ou ameaça de alto risco
- `medium` ou `fail-if:risk=mediumup` - Falha se for encontrada uma vulnerabilidade ou ameaça de risco Médio ou Alto
- `low` ou `fail-if:risk=lowup` - Falha se for encontrada uma vulnerabilidade ou ameaça de risco Baixo, Médio ou Alto

NOTAS ADICIONAIS

Se você receber o erro **Script.ps1 não pode ser carregado porque a execução de scripts está desativada neste sistema**, inicie o PowerShell com direitos administrativos e insira o seguinte comando para ativar scripts:

```
set-executionpolicy remotesigned
```

Se você receber o erro **O termo Start-CodeouDynamicScan não é um arquivo de script ou um programa operável**, você precisa carregar o módulo Syhunt no PowerShell. Entre o comando Import-Module Syhunt e chame a função de scan novamente.

CARREGANDO O SYHUNT AUTOMATICAMENTE

No Windows 10, o Syhunt é carregado automaticamente.

No Windows 7, você pode carregar manualmente o Syhunt toda vez chamando **Import-Module Syhunt** antes de chamar funções Syhunt ou você pode carregar o Syhunt automaticamente - adicione o Syhunt ao seu perfil do PowerShell e o Syhunt será carregado automaticamente:

1. Vá para o caminho do perfil do PowerShell, que por padrão é: C:\Documents and Settings\User\My Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1
2. Crie o arquivo **Microsoft.PowerShell_profile.ps1**
3. Adicione a linha **Import-Module Syhunt** e salve o arquivo.

Para documentação adicional do produto, visite **syhunt.com/docs/br**