

# Azure DevOps

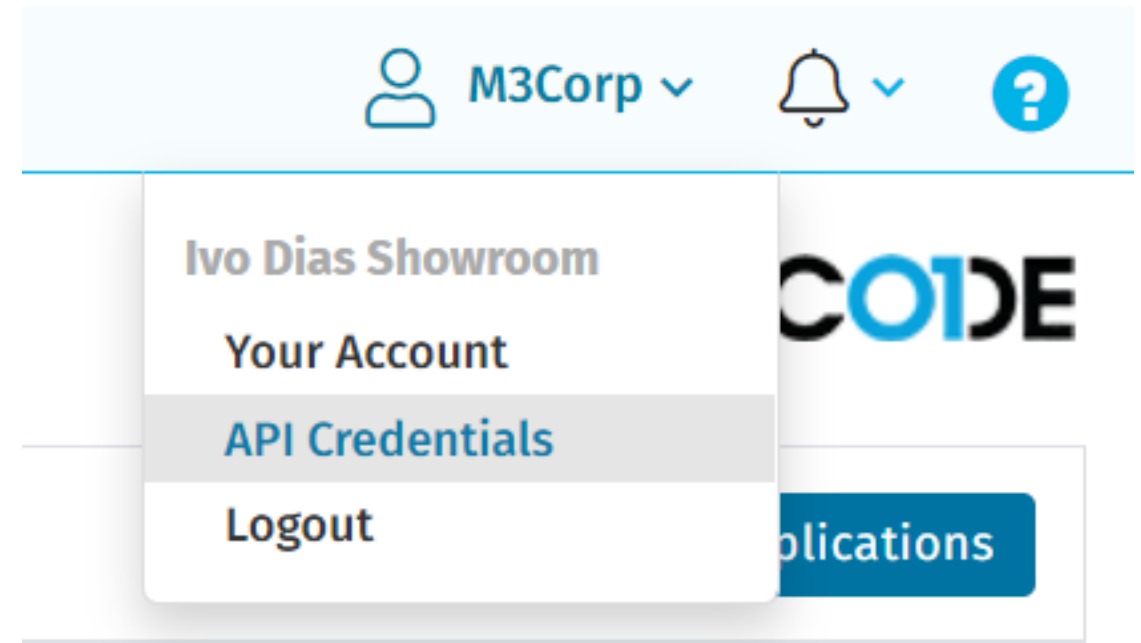
Guia de implementação

VERACODE



# Credenciais

- Nosso primeiro passo é obter as credenciais no portal da Veracode
- O recomendado é a criação de um usuário API específico para essas integrações
- Conforme a imagem ao lado, precisamos apenas clicar no nosso usuário no canto superior esquerdo e selecionamos a opção de credenciais



# Credenciais



- Vamos precisar do ID e da Secret Key para fazer as integrações
- Por padrão, essas credenciais duram 1 ano, mas conforme a imagem é possível revoga-las a qualquer momento

## Credentials Details

[Generate API Credentials](#)[Revoke API Credentials](#)

ID:

\*\*\*\*\*

Secret Key:

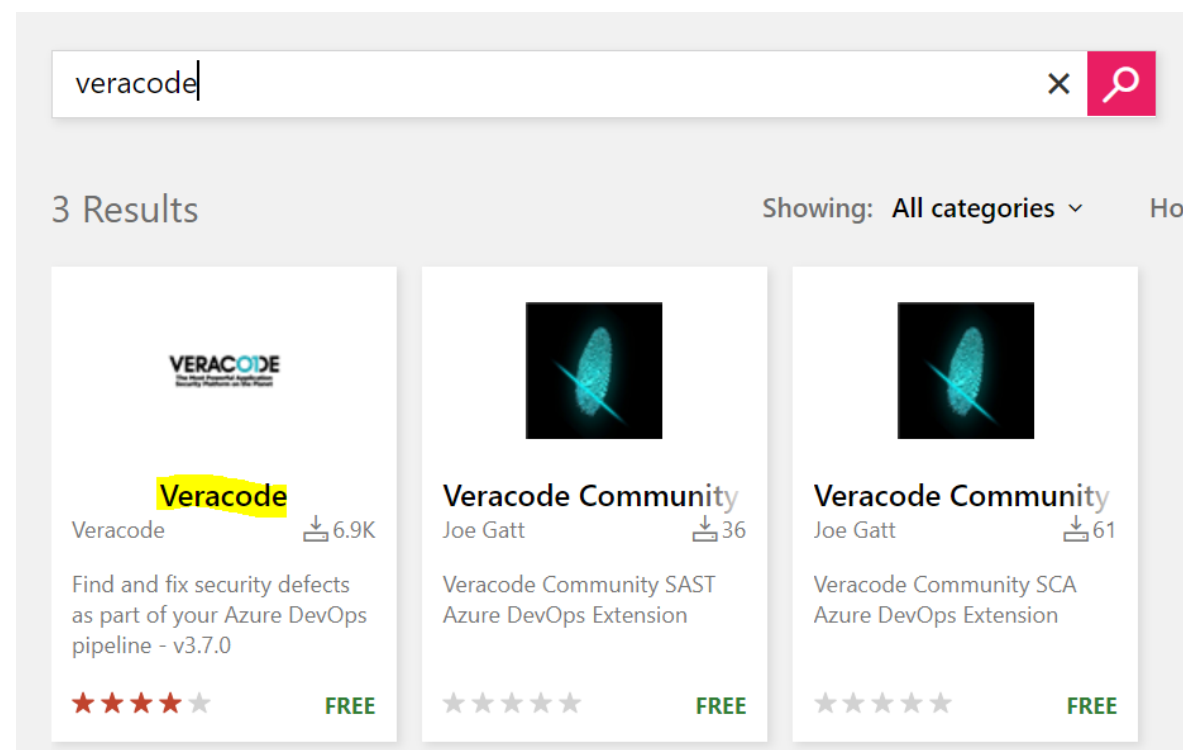
\*\*\*\*\*

**Created:** 13 Jul 2021 @ 9:21 am EDT

**Expires:** 13 Jul 2022 @ 9:21 am EDT

# Obtendo o Plug-in

- Dentro da loja de extensões, buscamos por “Veracode”
- A primeira opção é a do plug-in oficial
- Depois de instalarmos ela, vamos ter acesso a duas tarefas, que vamos ver no detalhe nos próximos slides
- É preciso que o agente tenha o Java configurado (nas imagens da Microsoft normalmente ele já vem disponível)



# Veracode Upload and Scan



- Essa é a Task que faz as análises, para fazer o login nela, precisamos informar os dados que pegamos no portal
- O ideal é a criação de uma [Service Connection](#) dentro do Azure, assim conseguimos tirar essa informação da tarefa e também aproveitar para uma implementação mais fácil em outros pipelines e tarefas

Veracode Upload and Scan ⓘ

[Link settings](#) [View YAML](#) [Remove](#)

Task version 3.\* ▼

Display name \*  
Upload and scan

Connection Details ^

Select Connection Source \* ⓘ

☒ Service Connection ☐ Credentials

Select Service Connection \* ⓘ | [Manage](#)

Base ▼ [Refresh](#) [+ New](#)

# Veracode Upload and Scan



- Para fazermos o scan precisamos de pouca coisa, basicamente apenas um nome de aplicação, um identificador para o scan (como um número de versão) e o caminho dos arquivos (conforme o [guia de empacotamento](#))
- Conforme a imagem, podemos ter todas essas informações abstraídas para ser aproveitadas em todos os processos
- Na parte dos resultados, temos uma checkbox para informar se queremos esperar a análise completar

Veracode Scan Settings ^

Application Name \* ⓘ

AzDevOps.\$(Build.DefinitionName)

Scan Name \* ⓘ

\$(build.buildNumber)

Filepath \* ⓘ

\$(Build.ArtifactStagingDirectory)/\$(Build.BuildId).zip

Advanced Scan Settings v

Veracode Scan Results ^

☒ Import Results upon Scan Completion ⓘ

☐ Fail build if application fails security policy ⓘ

Fail build if no scan results within (in minutes) \* ⓘ

360

# Veracode Upload and Scan



- Nas opções avançadas podemos fazer a implementação de [Sandbox](#)(relatório separado do principal)
- Um exemplo de uso é ter configurado para Stages específicos, como DEV e QA/HMG
- Uma opção muito importante é a da criação de perfil de aplicação, com ela ativa o sistema sempre verifica se já existe um com o nome informado e caso não tenha, já faz a criação automaticamente no portal
- Podemos também ativar a opção de “quebrar” o pipeline caso ocorra alguma falha no envio dos arquivos

## Advanced Scan Settings ^

Sandbox Name ⓘ

☐ Create Sandbox ⓘ

Optional Arguments ⓘ

☒ Create Application Profile ⓘ

☐ Fail build if Upload and Scan build step fails ⓘ

# Veracode Flaw Importer



- Essa tarefa é a responsável por fazer a importação das falhas encontradas para ao Azure Boards
- O processo de login é exatamente o mesmo da anterior
- Para utiliza-la, precisamos apenas informar qual a Aplicação que queremos importar. Com isso, conseguimos importar qualquer registro no portal, mesmo de análises feitas em outras ferramentas de CI/CD, e até mesmo iniciadas manualmente via portal ou linha de comando
- Em alguns casos, quando utilizamos templates diferentes do padrão, pode ser preciso uma [personalização](#)

Application Name \* ⓘ

AzDevOps.\$(Build.DefinitionName)

Sandbox Name ⓘ

Work Item Settings ^

Import \* ⓘ

All Flaws

Work Item Type \* ⓘ

Issue

Area \* ⓘ

\$(system.teamProject)

☒ Add CWE as a Tag ⓘ



# Veracode Flaw Importer



Essa é a visão que temos das falhas importadas com a tarefa

## Work items

Recently updated ▾ | + New Work Item ▾ | ↗ Open filtered view in Queries | Column Options ...

DVWA				Types ▾	Assigned to ▾	States ▾	Area ▾
ID	Title				Assigned To		State
803	Veracode Flaw (Static): Untrusted Initialization, PP_DEMO.PHP...	...	Unassigned				To Do
802	Veracode Flaw (Static): Untrusted Initialization, PP_DEMO.PHP.DVW...		Unassigned				To Do
801	Veracode Flaw (Static): Untrusted Initialization, PP_DEMO.PHP.DVW...		Unassigned				To Do
800	Veracode Flaw (Static): Untrusted Initialization, PP_DEMO.PHP.DVW...		Unassigned				To Do
799	Veracode Flaw (Static): Untrusted Initialization, PP_DEMO.PHP.DVW...		Unassigned				To Do

**ISSUE 803**  
803 Veracode Flaw (Static): Untrusted Initialization, PP\_DEMO.PHP.DVWA, Flaw 87

Unassigned [0 comments](#) [Build\\_12576517](#) [CWE\\_454](#) [+](#)

**State:** ☒ To Do **Area:** PP - DEMOs

**Reason:** Added to backlog **Iteration:** PP - DEMOs

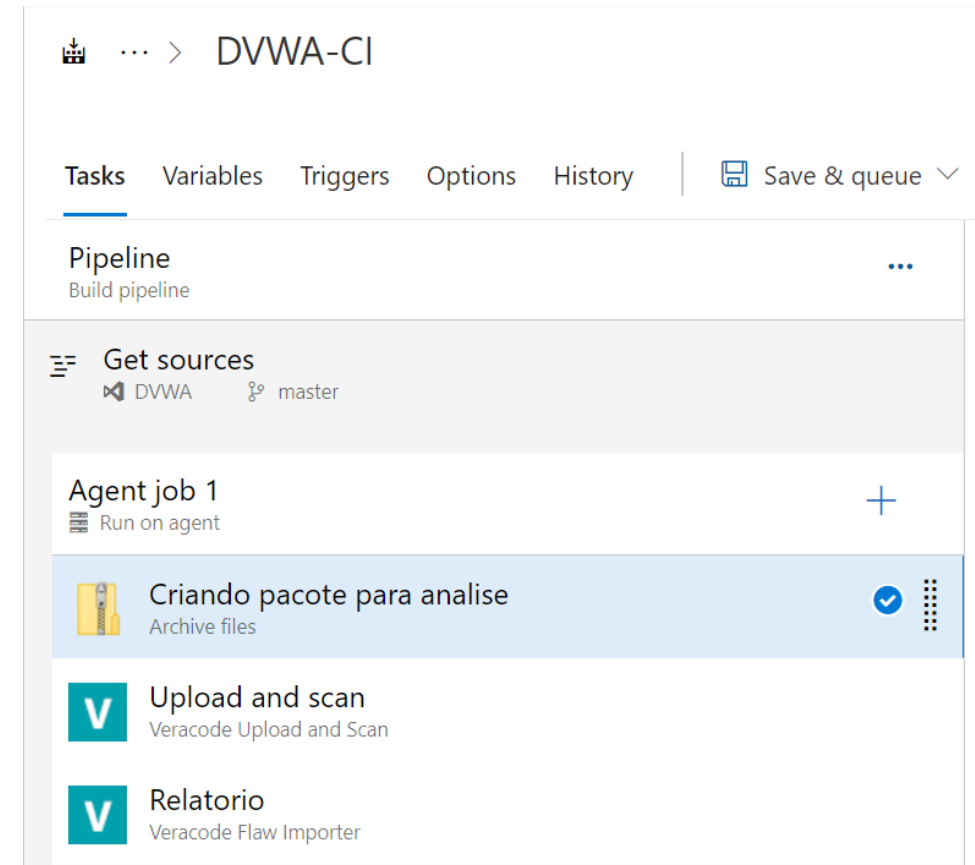
**Description**  
**Veracode Links :** [Application Policy Flaw](#)  
**CWE :** [454](#) External Initialization of Trusted Variables or Data Stores  
**Module :** PHP files within 254.zip  
**Source :** medium.php  
**Line Number :** 23  
**Attack Vector :** !php\_standard\_ns.shell\_exec  
**Description :**  
This call to !php\_standard\_ns.shell\_exec() invokes an external shell. Environment variables inherited from the calling program as well as those modified by the application itself will be passed to the shell. In light of the vulnerability in the shell bash described in CVE-2014-6271, the runtime environment should be reviewed to ensure that this is not an exploitable call.  
  
If this application is being run in an environment in which bash is present, ensure that the version of bash used always includes the most recent security updates. Also, in the application itself, consider creating an allowlist of environment variables to protect against external contamination.  
  
**References:**

**Planning**  
Priority  
2  
Effort

# Veracode - Pipeline Completo



- E pronto, esses são todos os passos necessários para implantarmos Veracode no Azure DevOps
- Conforme a imagem ao lado, um pipeline completo para a análise de um projeto em PHP tem apenas 3 tarefas
- Mesmo com um processo tão simples, com isso teremos acesso a uma análise completa do código, uma taxa mínima de falsos positivos, análise de componentes de terceiros e outras vantagens da Veracode, já com a importação de todas as informações para o Azure



Entre em contato com a nossa  
equipe para mais detalhes:  
[vendas@m3corp.com.br](mailto:vendas@m3corp.com.br)

SECURING THE SOFTWARE  
THAT POWERS YOUR WORLD

---

VERACODE