



m3corp

VERACODE

Linux e Mac

Guia de implementação



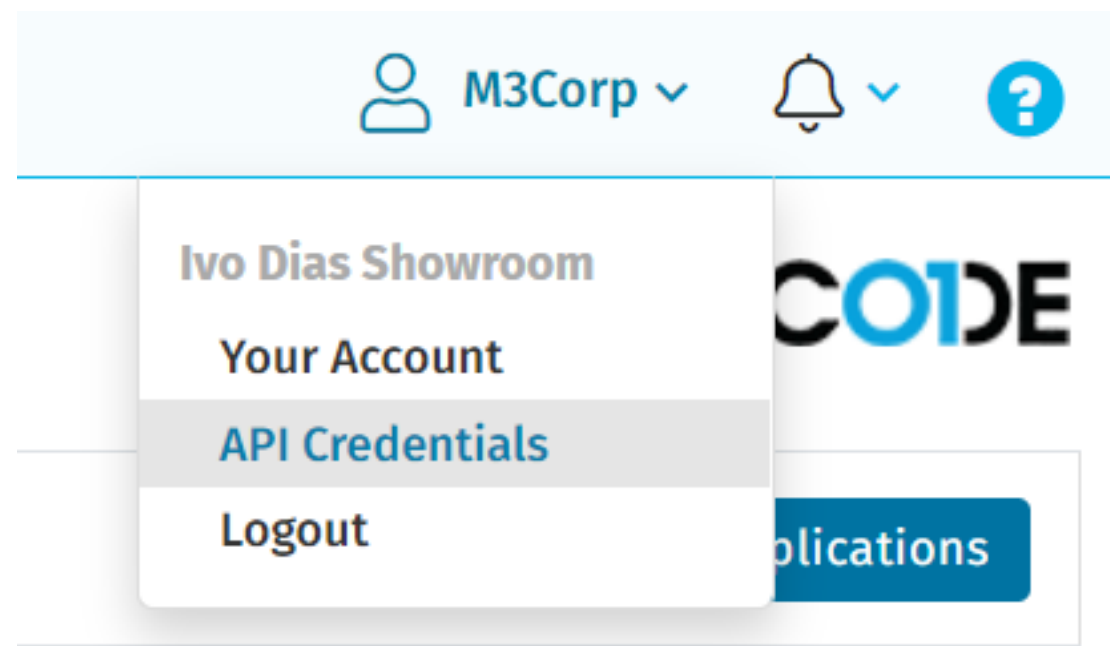
Considerações iniciais

- Para sistemas Mac e Linux, a implementação é a mesma devido ao processo ser feito utilizando Shell script
- Precisamos ter o Java configurado, já que vamos utilizar um .jar para fazer os scans
- Esse processo pode ser feito em qualquer ferramenta de CI/CD que permita instalar o Java e rodar comandos Shell
- O objetivo do material é mostrar como fazer as integrações, mas sem explicar muito sobre as ferramentas. Pode ficar tranquilo que também temos materiais explicando em detalhes cada uma delas.

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

Credenciais



- Para o SCA, precisamos criar um agente e um token para utilizarmos
- No portal, entramos na seção do SCA, clicamos em Agent-Based Scan, Actions e Create:

The screenshot shows the Veracode SCA portal. The top navigation bar includes a home icon, 'My Portfolio', 'Scans & Analysis' (selected), 'Analytics', 'Policies', and 'Security Training'. On the right, there's a user profile for 'M3Corp', a notification bell, and a help icon. Below the navigation bar, the main heading is 'Software Composition Analysis' with the Veracode logo. To the right of the heading are links for 'Vulnerability Database' and a 'Start a Scan' button. The 'Agent-Based Scan' tab is highlighted in green. Below this, there's a 'Workspace List' section with a search bar, a dropdown for '14 workspaces', and a '10' items per page selector. A green 'Actions' button is visible, and a 'Create Workspace' button is highlighted in yellow. At the bottom, a table header lists columns: 'Workspaces', 'Total Projects', 'Total Issues', 'Vulnerability Issues', 'Library Issues', 'License Issues', 'Last Scan', 'Teams', and 'Actions'.

Credenciais



- Definimos um nome para o nosso Workspace, que é onde vamos armazenar os resultados do Agente

Create Workspace [X]

Enter Workspace Name

Workspace Name (required)

Cancel More Options Create

Credenciais



- Na tela de configuração do agente, vamos ter um guia para todas as integrações possíveis, com todas as orientações necessárias

◀ Back to Workspace List

Workspaces +

AzDVWA ▼

Issues 128

Projects 1

Vulnerabilities 125

Libraries 4

Licenses 2

Manage Workspace ▼

Settings

Agents

Teams


Rules

Notifications

Set Up Scanner


Choose your operating system or CI to set up an agent.

OS X




OS X 10.8 (Mountain Lion) and later

LINUX




Recent versions of 64-bit Linux supported


WINDOWS





Windows 7 and later, PowerShell 3 and later


Integration Options


 Travis CI ➤


 Circle CI ➤


 Maven ➤

 Jenkins / Hudson ➤

 CodeShip ➤

 Gradle ➤

 Bitbucket Pipelines ➤

 Atlassian Bamboo ➤

SCA - Analise de código de terceiros



- Conseguimos com o SCA analisar não só os componentes de terceiros de um projeto, como também de um repositório em uma ferramenta de GIT e imagens Docker
- Temos o retorno com todas as informações disponíveis no próprio terminal
- Aqui vamos mostrar como fazer a implementação mais simples, mas na [documentação](#) temos detalhes sobre tudo o que pode fazer com ele
- Para a utilização do SCA precisamos apenas do “SRCCLR_API_TOKEN” configurado como variável de ambiente

SCA - Analise de código de terceiros



- Com apenas essa linha de comando, fazemos a análise e já deixamos configurado para fazer um commit com atualizações de versões propostas (sendo necessário configurar as credenciais de acesso ao repositório)

```
# SCA  
curl -sSL 'https://download.sourceclear.com/ci.sh' | bash -s - scan --update-advisor --pull-request
```

- Podemos passar como parâmetro um arquivo específico, diretório, site ou imagem Docker
- Nesse caso, fazemos a análise na pasta onde está linha foi executada

SCA - Analise de código de terceiros



- No terminal temos um relatório geral do que foi analisado

```
Summary Report
Scan ID                a1dcaeff-3085-42ed-88ab-e747cba646c8
Scan Date & Time       Jul 05 2021 03:11PM UTC
Account type           ENTERPRISE
Scan engine            3.7.40 (latest 3.7.40)
Analysis time          20 seconds
User                   vsts
Project                /home/vsts/work/1/s
Package Manager(s)     Composer

Open-Source Libraries
Total Libraries         4
Direct Libraries        2
Transitive Libraries    2
Vulnerable Libraries    2
Third Party Code        97.6%
```

SCA - Analise de código de terceiros



- No terminal temos um relatório geral do que foi analisado

Vulnerabilities - Public Data		
CVE-2019-3809	High Risk	Server-side Request Forgery (SSRF)
CVE-2015-5358	High Risk	Cross-Site Request Forgery(CSRF)
CVE-2018-10891	High Risk	Cross-Site Scripting (XSS)
CVE-2015-5332	High Risk	Denial Of Service (DoS) Through Disk
CVE-2017-2641	High Risk	SQL Injection
CVE-2014-7845	High Risk	Insecure Random Password Generation
CVE-2014-3541	High Risk	PHP Object Injection
CVE-2019-3850	Medium Risk	Information Disclosure
CVE-2014-0213	Medium Risk	Multiple Cross-Site Request Forgery (
CVE-2014-0216	Medium Risk	Information Disclosure
CVE-2014-3617	Medium Risk	Information Disclosure
CVE-2019-14881	Medium Risk	Cross-Site Scripting (XSS)
CVE-2018-1134	Medium Risk	Unauthorised Downloads
CVE-2021-20185	Medium Risk	Denial Of Service (DoS)
CVE-2018-1135	Medium Risk	Unauthorised Arbitrary File Downloads
CVE-2019-14884	Medium Risk	Cross-site Scripting (XSS)
CVE-2018-1136	Medium Risk	Unauthorised Editing To Web Pages
CVE-2019-3847	Medium Risk	Cross-Site Scripting (XSS)

SCA - Analise de código de terceiros



- No terminal temos um relatório geral do que foi analisado

Issues			
Issue ID	Issue Type	Severity	Description
80175234	Vulnerability	6.4	NO-CVE: Cross-site Scripting (XSS)
80176685	Vulnerability	7.5	CVE-2017-2641: SQL Injection
80176686	Vulnerability	7.5	CVE-2014-3541: PHP Object Injection
80176687	Vulnerability	7.5	CVE-2018-10891: Cross-Site Scripting (XSS)
80176688	Vulnerability	7.5	CVE-2014-7845: Insecure Random Password Generation
80176689	Vulnerability	7.5	CVE-2019-3809: Server-side Request Forgery (SSRF)
80176690	Vulnerability	7.1	CVE-2015-5332: Denial Of Service (DoS) Through Disk Cor
80176691	Vulnerability	7.1	CVE-2015-5358: Cross-Site Request Forgery(CSRF)
80176692	Vulnerability	6.8	CVE-2019-10186: Cross-site Scripting (XSS)
80176693	Vulnerability	6.8	CVE-2015-1493: Directory Traversal
80176694	Vulnerability	6.8	CVE-2015-0213: Cross-Site Request Forgery (CSRF)
80176695	Vulnerability	6.8	CVE-2015-0218: Denial Of Service (DoS) Through Cross-si
80176696	Vulnerability	6.8	CVE-2015-2268: Regular Expression Denial Of Service (Re
80176697	Vulnerability	6.8	CVE-2014-7838: Cross-site Request Forgery (CSRF)
80176698	Vulnerability	6.8	CVE-2018-16854: Cross-site Request Forgery (CSRF)
80176699	Vulnerability	6.8	CVE-2014-7836: Cross-site Request Forgery (CSRF)
80176700	Vulnerability	6.8	CVE-2016-2157: Cross-site Request Forgery (CSRF)
80176701	Vulnerability	6.8	CVE-2014-0214: Session Hijack

SCA - Analise de código de terceiros



- No terminal temos um relatório geral do que foi analisado

```
Update Advisor
Library Name & Version      Safe Version
moodle/moodle v2.6.2        v3.10.4
appserver-io/http 1.1.6     1.1.7

Full Report Details         https://sca.analysiscenter.veracode.com/teams/300u08
```

- Essa informação também fica disponível no portal da Veracode
- Como vemos nessa seção do Update Advisor, é proposta uma atualização para correção de bibliotecas e componentes com versões defasadas e com problemas



Pipeline Scan

- Para utilizarmos o Pipeline Scan de forma simples, precisamos apenas fazer o download do script e utiliza-lo
- Caso você tenha um repositório de arquivos, pode fazer o download apenas uma vez e disponibiliza-lo para ser utilizado pelos scripts
- No próximo slide vamos ter um exemplo da implementação mais simples, mas você também consegue fazer customizações no scan, conforme parâmetros disponíveis na [documentação](#)
- O ponto mais interessante da ferramenta é o seu retorno completo no terminal, podendo inclusive trabalhar no tratamento dessa resposta para análises personalizadas



Pipeline Scan

- Com essas três linhas de código, conseguimos fazer a configuração do script e fazer uma análise completa do projeto, com todos os benefícios de uma análise Veracode

```
# Pipeline Scan
curl -sSO https://downloads.veracode.com/securityscan/pipeline-scan-LATEST.zip
unzip pipeline-scan-LATEST.zip
java -jar pipeline-scan.jar -vid $veracodeID -vkey $veracodeAPIkey -f pacoteVeracode.zip
```

- Na primeira linha fazemos o download do JAR e na segunda descompactamos ele
- Na terceira fazemos a execução desse JAR passando como parâmetros as credenciais e o caminho do arquivo que queremos analisar

Pipeline Scan



- É assim que vemos o resultado no terminal, podendo ter ou não essa linha com a descrição da falha:

```
CWE-209: Information Exposure Through an Error Message: 1/s/external/phpids/0.6/lib/ids/caching/dat
Details: <span> The application calls the !operator_phpexit() function, which may expose informati
CWE-209: Information Exposure Through an Error Message: 1/s/external/phpids/0.6/lib/ids/caching/dat
Details: <span> The application calls the !operator_phpexit() function, which may expose informati
CWE-209: Information Exposure Through an Error Message: 1/s/external/phpids/0.6/lib/ids/caching/dat
Details: <span> The application calls the !operator_phpexit() function, which may expose informati
CWE-209: Information Exposure Through an Error Message: 1/s/external/phpids/0.6/docs/examples/examp
Details: <span> The application calls the !php_standard_ns.printf() function, which may expose info
CWE-209: Information Exposure Through an Error Message: 1/s/dvwa/includes/dvwaphpids.inc.php:97
Details: <span> The application calls the !php_standard_ns.printf() function, which may expose info
-----
Skipping 8 issues of Informational severity.
-----

=====
FAILURE: Found 93 issues!
=====
```




Wrapper API

- É nossa “estrela” para automações
- Permite que seja possível dentro da linha de comando o controle total da plataforma Veracode, não se limitando apenas a fazer scans, como também gerenciar permissões e times de usuários
- Recomendo a leitura da [documentação](#) para saber todas as possibilidades, já que nesse material veremos apenas como fazer um scan
- Sua implementação é fácil como a das outras ferramentas, e também pode ser utilizada no terminal
- Ao contrario das outras, não foi desenhada para um retorno tão detalhado no terminal, mas permite a geração de relatórios completos em PDF, XML e algumas outras opções

Wrapper API



- Fiz uma quebra de linhas para melhorar a visualização, mas podemos fazer toda a implementação com apenas 2

```
# Wrapper API
urlDownloadAPI="https://repo1.maven.org/maven2/com/veracode/vosp/api/wrappers/vosp-api-wrappers-java/20.12"
curl -L -o VeracodeJavaAPI.jar $urlDownloadAPI
java -jar VeracodeJavaAPI.jar
    -vid $veracodeID -vkey $veracodeAPIkey # Credenciais
    -action uploadandscan # Ação para fazer todo o processo de análise
    -appname "$appName" # Nome do perfil de aplicação
    -filepath "$zipArquivo" # Caminho do arquivo que vai ser analisado
    -version $numeroVersao # Numero de versão para identificação desse scan (ex. numero de build)
    -createprofile true # Cria automaticamente um perfil de App caso não exista
    -scantimeout 60 # Define um limite em minutos para o scan
```



Wrapper API

- Por mais que ela não seja desenhada para gerar retornos no terminal, como conversamos podemos obter todos os dados com um XML e trabalhar com esses retornos, inclusive para criar relatórios personalizados ou automações para envio de relatórios completos em PDF (com imagens e gráficos) por e-mail
- Pode ser utilizada com as ferramentas anteriores, para inventariar os perfis de aplicações que possui e até mesmo para criação e configuração de usuários
- Tudo isso é possível graças a um conjunto de ações, que podem ser [consultadas nesse material](#)



Conclusão

- Como vimos nesse material, mesmo a forma “difícil” de implementar Veracode é muito simples e propicia uma grande liberdade para sua equipe definir como e onde quer trabalhar
- Nossas ferramentas são desenhadas para uma utilização simples, mas sempre pensando em maximizar os resultados. Mesmo com uma ou duas linhas de código, vai conseguir uma análise completa do seu projeto, com a menor taxa de falsos positivos do mercado e de forma simultânea, sem precisar em se preocupar com infraestrutura, já que todos os scans são feitos em nossa nuvem

Entre em contato com a nossa
equipe para mais detalhes:
vendas@m3corp.com.br

SECURING THE SOFTWARE
THAT POWERS YOUR WORLD

VERACODE