Giriş

Tahmini gerekli süre: 20-25 dakika

Yazılım Bileşimi Analizi (SCA) için uygulamalı laboratuvara hoş geldiniz.

Öğrenme Hedefleri

Bu laboratuvarı tamamladıktan sonra şunları yapabileceksiniz:

- **OWASP SCA Aracını** indirin ve kurun
- SCA aracını kullanarak bir projenin bileşenlerindeki güvenlik açıklarını tespit edin
- Tarama sonuçlarını JSON ve HTML formatlarında çıktı alın
- Aracın çıktısını nasıl analiz edeceğinizi anlayın
- Raporların nerede saklandığını keşfedin

Yazılım Bileşimi Analizi (SCA) Nedir?

Modern uygulamalar, üçüncü taraf ve açık kaynak bileşenler kullanılarak geliştirilebilir. Bu bileşenleri diğer insanlar kodlar. Bileşenlerin güvenli olduğunu nasıl bilirsiniz? Güvenliklerini nasıl doğrulayabilirsiniz? Bunu, uygulamalarınızda kullanılan bileşenleri incelemek için yazılım bileşimi analiz araçlarını nasıl kullanacağınızı öğrenerek yapabilirsiniz. Kodunuzun bilinen bileşen güvenlik açıklarından korunduğundan emin olabilirsiniz.

Yazılım Bileşimi Analizi (SCA)

Yazılım Bileşimi Analizi (SCA), uygulama geliştirme sırasında üçüncü taraf ve açık kaynak bileşenlerin kullanımından kaynaklanan risk alanlarını tanımlama sürecidir.

SCA araçları, aşağıdaki gibi birkaç risk faktörünü belirleyebilir:

- Eski bileşenler.
- Bilinen güvenlik açıklarına sahip bileşenler.
- Bileşen kalitesi.
  - Güvenlik açısından, bir bileşen kötü bakım görüyorsa veya çok küçük bir topluluk tarafından destekleniyorsa daha düşük kaliteli olarak değerlendirilebilir.
- Geçişli bağımlılıklar.
  - SCA araçları, **geçişli bağımlılıklardaki** güvenlik açıklarını takip edebilir. Bir bileşen başka bir bileşene bağımlı olduğunda, o bağımlılığa **geçişli** denir.
- Dış hizmetler
  - Bir bileşen, Web API'leri gibi dış hizmetlerle etkileşimde bulunabilir. SCA araçları, bu etkileşimin bir güvenlik açığı olabileceğini belirleyebilir.

Dependency-Check, bir projenin bağımlılıkları içinde yer alan kamuya açık olarak duyurulmuş güvenlik açıklarını tespit etmeye çalışan bir Yazılım Bileşimi Analizi (SCA) aracıdır. Bağımlılıklar, kodunuzun ek işlevsellik için güvendiği yazılım bileşenleridir. SCA aracı, bağımlılığı, belirlenen Common Platform Enumeration (CPE) tanımlayıcılarını ve ilgili Common Vulnerability and Exposure (CVE) girişlerini listeleyen bir rapor oluşturacaktır.

Bu uygulamalı laboratuvar çalışmasında, **OWASP SCA Dependency-checker** aracının kullanımını keşfedeceğiz.

Adım 1: OWASP SCA Araçını Kurma

Uygulamalı laboratuvara başlamadan önce biraz hazırlık yapmanız gerekiyor.

Göreviniz

1. OWASP bağımlılık kontrolü betiğini indirmek ve kurmak için `wget` komutunu çalıştırın:

```
wget -O dependency-check.zip https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-CD0267EN-SkillsNetwork/labs/module2/data/dependency-check.zip && unzip dependency-check.zip && chmod +x
```

Sonuçlar

Aşağıdakine benzer bir çıktı görmelisiniz:

```
Problems        theia@theia-samaahs: /home/project ×

theia@theia-samaahs:/home/project$ wget -O dependency-check.zip https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud
/IBM-CD0267EN-SkillsNetwork/labs/module2/data/dependency-check.zip && unzip dependency-check.zip && chmod +x dependency-check/bin/
dependency-check.sh && sudo echo "alias dependency-check=$(pwd)/dependency-check/bin/dependency-check.sh" >> ~/.bashrc && source ~
/.bashrc
--2022-09-14 08:08:45--  https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-CD0267EN-SkillsNetwork/labs/module
2/data/dependency-check.zip
Resolving cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud (cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud).
.. 169.63.118.104
Connecting to cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud (cf-courses-data.s3.us.cloud-object-storage.appdomain.clo
ud)|169.63.118.104|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 24493843 (23M) [application/zip]
Saving to: 'dependency-check.zip'

dependency-check.zip             100%[=================================================>]  23.36M  45.1MB/s    in 0.5s

2022-09-14 08:08:46 (45.1 MB/s) - 'dependency-check.zip' saved [24493843/24493843]

Archive:  dependency-check.zip
   creating: dependency-check/
   creating: dependency-check/bin/
   creating: dependency-check/lib/
   creating: dependency-check/plugins/
   creating: dependency-check/licenses/
   creating: dependency-check/licenses/commons-cli/
  inflating: dependency-check/bin/completion-for-dependency-check.sh
  inflating: dependency-check/bin/dependency-check.sh
  inflating: dependency-check/bin/dependency-check.bat
  inflating: dependency-check/lib/aho-corasick-double-array-trie-1.2.3.jar
  inflating: dependency-check/lib/android-json-0.0.20131108.vaadin1.jar
  inflating: dependency-check/lib/annotations-23.0.0.jar
  inflating: dependency-check/lib/ant-1.10.12.jar
  inflating: dependency-check/lib/checker-qual-3.12.0.jar
  inflating: dependency-check/lib/commons-beanutils-1.9.4.jar
  inflating: dependency-check/lib/commons-cli-1.5.0.jar
  inflating: dependency-check/lib/commons-codec-1.15.jar
  inflating: dependency-check/lib/commons-collections-3.2.2.jar
  inflating: dependency-check/lib/commons-collections4-4.4.jar
  inflating: dependency-check/lib/commons-compress-1.21.jar
  inflating: dependency-check/lib/commons-dbcp2-2.9.0.jar
  inflating: dependency-check/lib/commons-digester-2.1.jar
  inflating: dependency-check/lib/commons-io-2.11.0.jar
  inflating: dependency-check/lib/commons-jcs-core-2.2.1.jar
  inflating: dependency-check/lib/commons-lang3-3.12.0.jar
  inflating: dependency-check/lib/commons-logging-1.2.jar
  inflating: dependency-check/lib/commons-pool2-2.10.0.jar
  inflating: dependency-check/lib/commons-text-1.9.jar
  inflating: dependency-check/lib/commons-validator-1.7.jar
  inflating: dependency-check/lib/compiler-0.9.6.jar
  inflating: dependency-check/lib/cpe-parser-2.0.2.jar
  inflating: dependency-check/lib/dependency-check-cli-7.1.1.jar
  inflating: dependency-check/lib/dependency-check-core-7.1.1.jar
  inflating: dependency-check/lib/dependency-check-utils-7.1.1.jar
  inflating: dependency-check/lib/error_prone_annotations-2.11.0.jar
  inflating: dependency-check/lib/failureaccess-1.0.1.jar
  inflating: dependency-check/lib/gson-2.8.5.jar
  inflating: dependency-check/lib/guava-31.1-jre.jar
  inflating: dependency-check/lib/h2-2.1.210.jar
  inflating: dependency-check/lib/j2objc-annotations-1.3.jar
  inflating: dependency-check/lib/jackson-annotations-2.13.3.jar
  inflating: dependency-check/lib/jackson-core-2.13.3.jar
  inflating: dependency-check/lib/jackson-databind-2.13.3.jar
  inflating: dependency-check/lib/jackson-module-afterburner-2.13.3.jar
  inflating: dependency-check/lib/javax.inject-1.jar
  inflating: dependency-check/lib/javax.json-1.1.4.jar
  inflating: dependency-check/lib/javax.ws.rs-api-2.0.1.jar
  inflating: dependency-check/lib/jcl-over-slf4j-1.7.28.jar
  inflating: dependency-check/lib/jdiagnostics-1.0.7.jar
  inflating: dependency-check/lib/joda-time-2.10.4.jar
  inflating: dependency-check/lib/jsoup-1.15.1.jar
  inflating: dependency-check/lib/jsr305-3.0.2.jar
  inflating: dependency-check/lib/listenablefuture-9999.0-empty-to-avoid-conflict-with-guava.jar
  inflating: dependency-check/lib/logback-classic-1.2.11.jar
  inflating: dependency-check/lib/logback-core-1.2.11.jar
  inflating: dependency-check/lib/lucene-analyzers-common-8.11.1.jar
  inflating: dependency-check/lib/lucene-core-8.11.1.jar
  inflating: dependency-check/lib/lucene-queries-8.11.1.jar
  inflating: dependency-check/lib/lucene-queryparser-8.11.1.jar
  inflating: dependency-check/lib/lucene-sandbox-8.11.1.jar
  inflating: dependency-check/lib/minlog-1.3.1.jar
  inflating: dependency-check/lib/ossindex-service-api-1.8.1.jar
  inflating: dependency-check/lib/ossindex-service-client-1.8.1.jar
  inflating: dependency-check/lib/package-url-java-1.1.1.jar
  inflating: dependency-check/lib/packager-core-0.18.0.jar
  inflating: dependency-check/lib/packager-rpm-0.18.0.jar
  inflating: dependency-check/lib/packageurl-java-1.4.1.jar
  inflating: dependency-check/lib/pecoff4j-0.0.2.1.jar
  inflating: dependency-check/lib/retirejs-core-3.0.3.jar
  inflating: dependency-check/lib/semver4j-3.1.0.jar
  inflating: dependency-check/lib/slf4j-api-1.7.36.jar
  inflating: dependency-check/lib/spotbugs-annotations-4.7.0.jar
  inflating: dependency-check/lib/toml4j-0.7.2.jar
  inflating: dependency-check/lib/velocity-engine-core-2.3.jar
  inflating: dependency-check/lib/xz-1.8.jar
  inflating: dependency-check/LICENSE.txt
  inflating: dependency-check/NOTICE.txt
  inflating: dependency-check/licenses/commons-cli/LICENSE.txt
  inflating: dependency-check/README.md
theia@theia-samaahs:/home/project$
```

Artık komut satırında `dependency-check` kullanarak OWASP SCA aracını çalıştırabileceksiniz.

Adım 2: Kaynak Kodunu İndirin

Sonraki adımda taramak için bazı kaynak kodlarına ihtiyacımız var. Güvenlik eğitimi amaçları için özel olarak oluşturulmuş **OWASP Juice Shop** adlı popüler bir zayıf uygulama örneğini kullanacağız.

Göreviniz

1. Kaynak kodunu indirmek için aşağıdaki `git clone` komutunu çalıştırın:

```
git clone https://github.com/juice-shop/juice-shop.git
```

Sonuçlar

Çıktınız aşağıdaki gibi görünmelidir:

```
theia@theia-samaahs:/home/project$ git clone https://github.com/juice-shop/juice-shop.git
Cloning into 'juice-shop'...
remote: Enumerating objects: 115562, done.
remote: Counting objects: 100% (57/57), done.
remote: Compressing objects: 100% (44/44), done.
remote: Total 115562 (delta 12), reused 52 (delta 11), pack-reused 115505
Receiving objects: 100% (115562/115562), 179.76 MiB | 34.31 MiB/s, done.
Resolving deltas: 100% (89637/89637), done.
theia@theia-samaahs:/home/project$
```

2. 'uploads' klasörünü ve tüm içeriğini silmek için aşağıdaki komutu çalıştırın:

```
rm -rf juice-shop/frontend/src/assets/public/images/uploads/
```

Sonuçlar

```
theia@theia-daniak:/home/project$ rm -rf juice-shop/frontend/src/assets/
public/images/uploads/
theia@theia-daniak:/home/project$
```

Artık **Juice Shop** uygulamasını analiz etmeye hazırsınız.

Adım 3: Juice Shop Bileşenlerinde SCA Çalıştırma

OWASP SCA aracı, Birleşik İşaretleme Dili (HTML) raporu veya JavaScript Nesne Notasyonu (JSON) çıktısı oluşturabilir. Bu adımda, aracı çalıştıracağız ve sonuçları JSON çıktısı olarak alacağız.

Göreviniz

1. Aşağıdaki seçeneklerle **Juice Shop** kaynak kodunda `dependency-check` komutunu kullanarak JSON çıktısı üretin:

```
dependency-check -f JSON --prettyPrint --scan juice-shop
```

Not: Bu biraz zaman alabilir çünkü ilk kez çalıştırdığınızda, CVE veritabanından tüm güncellemeleri indirmesi gerekir. Sonraki çalıştırmalar daha hızlı tamamlanacaktır.

Bu komut, OWASP SCA Aracı tarafından bulunan herhangi bir savunmasız bileşen hakkında bilgi içerebilecek `dependency-check-report.json` adlı bir dosya üretecektir.

Tarama Sonuçları

Sonuçlar

Tarama sonuçları buna benzer görünecektir:

```
Terminal  Help

Problems   theia@theia-samaahs: /home/project/juice-shop  ×

Dependency-Check is an open source tool performing a best effort analysis of 3rd party dependenc
ies; false positives and false negatives may exist in the analysis performed by the tool. Use of
 the tool and the reporting provided constitutes acceptance for use in an AS IS condition, and t
here are NO warranties, implied or otherwise, with regard to the analysis or its use. Any use of
 the tool and the reporting provided is at the user's risk. In no event shall the copyright hold
er or OWASP be held liable for any damages whatsoever arising out of or in connection with the u
se of this tool, the analysis performed, or the resulting report.


   About ODC: https://jeremylong.github.io/DependencyCheck/general/internals.html
   False Positives: https://jeremylong.github.io/DependencyCheck/general/suppression.html

💖Sponsor: https://github.com/sponsors/jeremylong


[INFO] Analysis Started
[WARN] Exception extracting archive 'arbitraryFileWrite.zip'.
[WARN] Exception extracting archive 'videoExploit.zip'.
[INFO] Finished Archive Analyzer (0 seconds)
[INFO] Finished File Name Analyzer (0 seconds)
[WARN] An error occurred with the .NET AssemblyAnalyzer, please see the log for more details.
[ERROR] Exception occurred initializing Assembly Analyzer.
[WARN] No lock file exists - this will result in false negatives; please run `npm install --pack
age-lock`
[WARN] No lock file exists - this will result in false negatives; please run `npm install --pack
age-lock`
[WARN] Analyzing `/home/project/juice-shop/package.json` - however, the node_modules directory d
oes not exist. Please run `npm install` prior to running dependency-check
[WARN] Analyzing `/home/project/juice-shop/frontend/package.json` - however, the node_modules di
rectory does not exist. Please run `npm install` prior to running dependency-check
[INFO] Finished Node.js Package Analyzer (0 seconds)
[INFO] Finished Dependency Merging Analyzer (0 seconds)
[INFO] Finished Version Filter Analyzer (0 seconds)
[INFO] Finished Hint Analyzer (0 seconds)
[INFO] Created CPE Index (3 seconds)
[INFO] Finished CPE Analyzer (4 seconds)
[INFO] Finished False Positive Analyzer (0 seconds)
[INFO] Finished NVD CVE Analyzer (0 seconds)
[INFO] Finished RetireJS Analyzer (0 seconds)
[WARN] Unable to determine Package-URL identifiers for 20 dependencies
[INFO] Finished Sonatype OSS Index Analyzer (0 seconds)
[INFO] Finished Vulnerability Suppression Analyzer (0 seconds)
[INFO] Finished Dependency Bundling Analyzer (0 seconds)
[INFO] Analysis Complete (5 seconds)
[INFO] Writing report to: /home/project/juice-shop/./dependency-check-report.json
[ERROR] Could not execute .NET AssemblyAnalyzer
theia@theia-samaahs:/home/project/juice-shop$
```

Adım 4: JSON Sonuçlarını Analiz Etme

Adım 3'te komutu çalıştırdığımızda, rapor `dependency-check-report.json` dosyası `/home/project` klasöründe kaydedildi. Dosyayı manuel olarak açarak veya bu düğmeye basarak görüntüleyebilirsiniz:

Open **dependency-check-report.json** in IDE

JSON okunması zor olabileceğinden, çıktıyı yalnızca etkilenen dosyaların adını gösterecek şekilde değiştirmek isteyebilirsiniz.

Bunu yapmak için `jq` komutunu kullanabilir ve JSON çıktısını yalnızca etkilenen dosyaların yolunu gösterecek şekilde filtrelemek için `'.dependencies[].filePath'` parametresini ekleyebilirsiniz.

Göreviniz

1. OWASP aracının bulduğu güvenlik açıklarına sahip yazılım bileşenlerini listelemek için aşağıdaki `jq` komutunu kullanın:

```
jq '.dependencies[].filePath' dependency-check-report.json
```

Not: Bu arama, `dependency-check-report.json` dosyasındaki `dependencies[]` dizisindeki tüm öğelerin `filePath` niteliğini döndürecektir.

Filtre Çıktı Sonuçları

Sonuçlar

Çıktı aşağıdaki gibi görünmelidir:

```
theia@theia-samaahs:/home/project/juice-shop$ jq '.dependencies[].filePath' dependency-check-rep
ort.json
"/home/project/juice-shop/.eslintrc.js"
"/home/project/juice-shop/frontend/.eslintrc.js"
"/home/project/juice-shop/frontend/.stylelintrc.js"
"/home/project/juice-shop/frontend/src/assets/private/CopyShader.js"
"/home/project/juice-shop/frontend/src/assets/private/EffectComposer.js"
"/home/project/juice-shop/Gruntfile.js"
"/home/project/juice-shop/frontend/src/assets/private/MaskPass.js"
"/home/project/juice-shop/frontend/src/assets/private/OrbitControls.js"
"/home/project/juice-shop/frontend/src/assets/private/RenderPass.js"
"/home/project/juice-shop/frontend/src/assets/private/ShaderPass.js"
"/home/project/juice-shop/frontend/src/assets/private/dat.gui.min.js"
"/home/project/juice-shop/test/files/invalidTypeForClient.exe"
"/home/project/juice-shop/frontend/src/karma.conf.js"
"/home/project/juice-shop/frontend/package.json"
"/home/project/juice-shop/package.json"
"/home/project/juice-shop/protractor.conf.js"
"/home/project/juice-shop/protractor.subfolder.conf.js"
"/home/project/juice-shop/frontend/src/assets/private/stats.min.js"
"/home/project/juice-shop/views/themes/themes.js"
"/home/project/juice-shop/frontend/src/assets/private/three.js"
theia@theia-samaahs:/home/project/juice-shop$
```

Güvenli bir Yazılım Geliştirme Yaşam Döngüsü (SDLC) içinde, her bileşen güvenlik açısından dikkatlice kontrol edilmeli ve doğrulanmalıdır. Zayıf olabilecek herhangi bir bağımlılık yükseltilmeli veya değiştirilmelidir.

Adım 5: HTML Raporu Oluşturma

JSON formatında çıktı verilen raporlar okunması zor olabilir. Bunun yerine tarama çıktınızı bir HTML raporuna göndermek daha uygun olabilir. HTML raporları bizim için daha kolay okunur ve yorumlanır.

Göreviniz

1. `juice-shop` klasörünün `--scan`'inden bir HTML raporu oluşturmak için `dependency-check` komutunu kullanın:

```
dependency-check --scan juice-shop
```

İşlem tamamlandığında, dosya gezgininizde `dependency-check-report.html` adında bir rapor bulunacaktır. Ne yazık ki, Cloud IDE ortamı bu HTML dosyasını görüntülemek için bir yol sunmamaktadır, bu nedenle raporu indirip web tarayıcınızda görüntülemeniz gerekecektir.

Hadi bir örnek HTML raporuna daha yakından bakalım.

Sonuçlar

Raporun **Proje:** bölümü **Tarama Bilgileri**ni gösterir. Bu bölüm, kullanılan bağımlılık denetleyicisinin sürümünü, raporun çalıştırıldığı tarih ve saati, taranan bağımlılık sayısı hakkında bazı ek bilgileri ve bulunan zafiyet sayısını içerir.

Raporun **Özet** bölümü bağımlılıkları isimlerine göre, Zafiyet ID'sine ve paket adına göre gösterir. Ayrıca her zafiyetin kritiklik seviyesini ve **Juice Shop** projesiyle ilgili diğer detayları da gösterir.

Dependency-Check is an open source tool performing a best effort analysis of 3rd party dependencies; false positives and false negatives may exist in the analysis performed by the tool. Use of the tool and the reporting provided constitutes whatsoever arising out of or in connection with the use of this tool, the analysis performed, or the resulting report.

How to read the report | **Suppressing false positives** | **Getting Help: github issues**

Sponsor

Project:

Scan Information (show all):

- *dependency-check version*: 7.1.1
- *Report Generated On*: Wed, 20 Jul 2022 21:09:42 GMT
- *Dependencies Scanned*: 15675 (10954 unique)
- *Vulnerable Dependencies*: 6
- *Vulnerabilities Found*: 11
- *Vulnerabilities Suppressed*: 0
- ...

Summary

Display: Showing Vulnerable Dependencies (click to show all)

Dependency	Vulnerability IDs	Package	Highest Severity	CVE Count	Confidence	Evidence
express-jwt:0.1.3	cpe:2.3:a:auth0:express-jwt:0.1.3:*:*:*:*:*:*:*	pkg:npm/express-jwt@0.1.3	CRITICAL	1	Highest	10
hbs:4.2.0	cpe:2.3:a:hbs_project:hbs:4.2.0:*:*:*:*:*:*:*	pkg:npm/hbs@4.2.0	MEDIUM	1	Highest	10
jquery.js		pkg:javascript/jquery@2.1.0	MEDIUM	4		3
jsonwebtoken:0.4.0	cpe:2.3:a:auth0:jsonwebtoken:0.4.0:*:*:*:*:*:*:*	pkg:npm/jsonwebtoken@0.4.0	CRITICAL	1	Highest	8
notevil:1.3.3	cpe:2.3:a:notevil_project:notevil:1.3.3:*:*:*:*:*:*:*	pkg:npm/notevil@1.3.3	MEDIUM	1	Highest	8
sanitize-html:1.4.2	cpe:2.3:a:punkave:sanitize-html:1.4.2:*:*:*:*:*:*:*	pkg:npm/sanitize-html@1.4.2	MEDIUM	3	Highest	8

Dependencies

express-jwt:0.1.3

Description:

JWT authentication middleware.

File Path: /home/project/juice-shop_14.1.1/package.json?/express-jwt:0.1.3

Referenced In Project/Scope:juice-shop:14.1.1

Evidence

Identifiers

- pkg:npm/express-jwt@0.1.3 (*Confidence*:Highest)
- cpe:2.3:a:auth0:express-jwt:0.1.3:*:*:*:*:*:*:* (*Confidence*:Highest) suppress

Gördüğünüz gibi, html versiyonu daha kolay yorumlanabilir. Tüm zafiyetlere tıklayarak onları anlamak ve tek tek düzeltmek için işlem yapabilirsiniz.

Sonuç

Tebrikler! **Yazılım Bileşimi Analizi (SCA)**'nın kodunuzun bağımlı olduğu herhangi bir bağımlılıkta bilinen güvenlik açıklarını nasıl tespit edebileceğinizi öğrendiniz. Bu, güvenli yazılım uygulama geliştirme sürecinin ayrılmaz bir parçasıdır.

Bu laboratuvar çalışmasında, **OWASP SCA Dependency-check Tool**'u indirip kurmayı ve yazılım uygulamasında kullanılan bileşenler üzerinde bağımlılık analizi yapmayı öğrendiniz. Tarama sonuçlarını JSON ve HTML formatlarında nasıl çıktı alacağınızı ve raporların nerede saklandığını öğrendiniz. HTML raporunun düzenine aşina oldunuz ve sonuçları nasıl analiz edeceğinizi öğrendiniz.

Sonraki Adımlar

Farklı güvenlik açıklarını tespit etmek, güvenli uygulama geliştirmede atılacak ilk adımlardan biridir. Bu güvenlik açıklarının arkasındaki anlamı anlamak, bunları azaltmak için düzeltici önlemler almak açısından faydalıdır. Öğrenmenin en iyi yolu, uygulamalı öğrenmektir.

Bir sonraki zorluğunuz, geliştirme ortamınıza OWASP SCA Dependency-Check kurmak, kodunuz üzerinde bağımlılık taramaları yapmak ve bulduğu sorunları düzeltmektir. Daha güvenli kod yazma yolunda iyi bir ilerleme kaydettiniz!

Author(s)

Sam Prokopchuk

Other Contributor(s)

Samaah Sarang
John J. Rofrano
Michelle R. Sanchez, 25 yılı aşkın kurumsal düzeyde teknik destek ve kurumsal teknik eğitim deneyimi olan Skill-up Technologies'de Eğitim Tasarımcısı.