Secure DevOps

Automated Mobile App Security Scanning

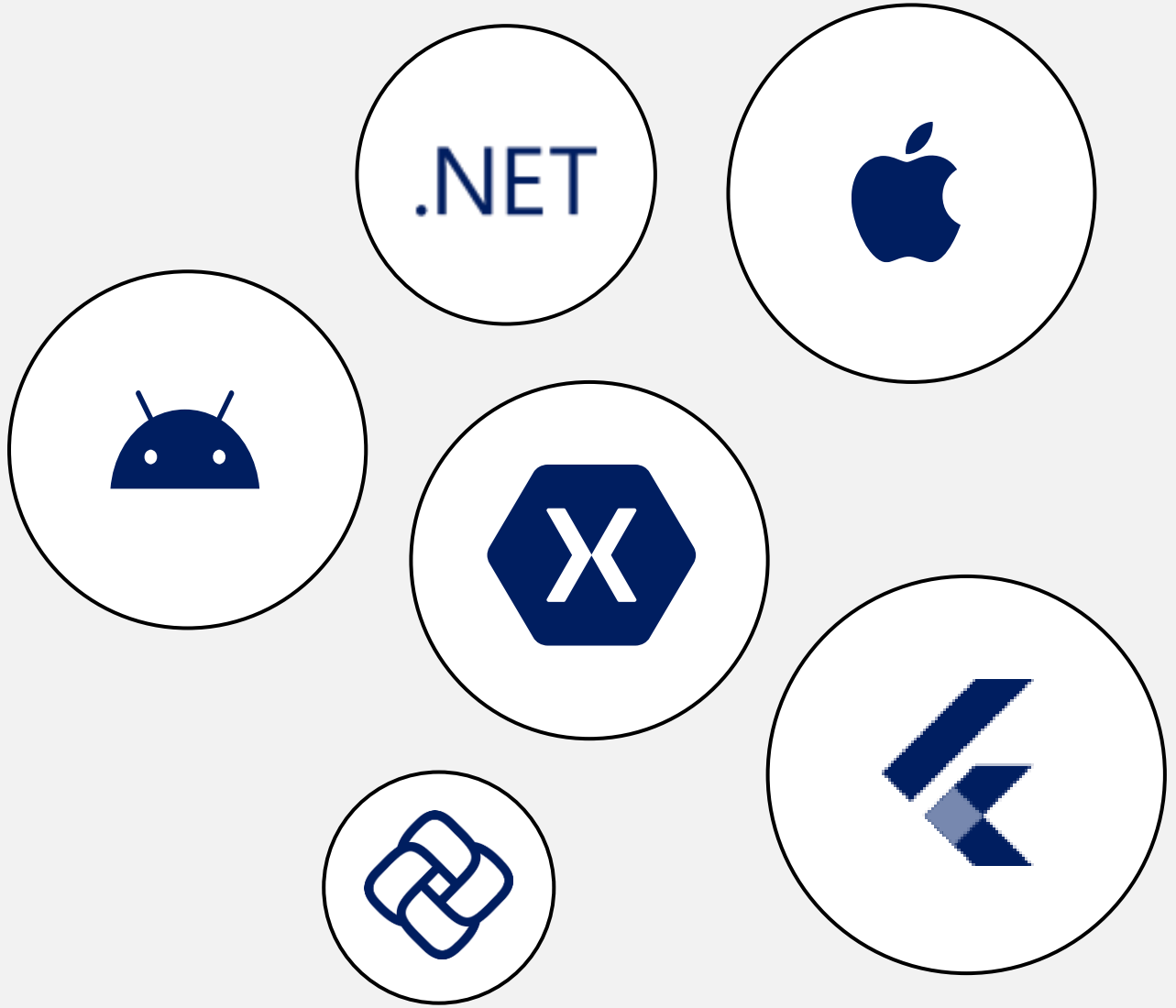
Feedback:

ConFoo 2025

Lester Botello

4 years at nventive

- Native / Mobile .NET / Flutter
- DevOps / Mobile Security
- Development Team Lead



DevSecOps

DevSecOps

- “Shift Left” mantra
- Security education
- DevSecOps process traits:
  - Traceability
  - Auditability
  - Visibility

SAST vs. DAST

SAST

Static Application Security Testing

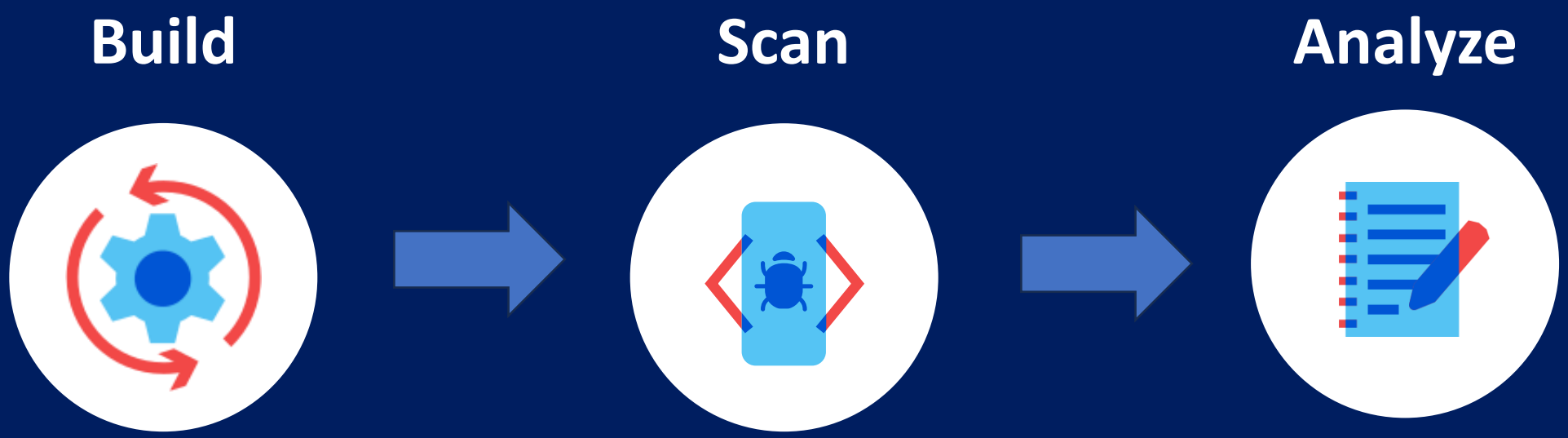
- Static analysis of source code or binaries.
- Early in the development lifecycle
- Focuses on the internal code structure
- Finds code-level issues early
- Cannot detect runtime issues

DAST

Dynamic Application Security Testing

- Dynamic analysis of the running application.
- Later in the development lifecycle
- Focuses on the external behavior of the app
- Detects runtime and environment issues.
- Cannot pinpoint exact code vulnerabilities

SAST in action



Static Analysis

Why automate static analysis for your app?

Ensure user data is protected

- Personally identifiable information (PII)
- Security information (passwords, biometrics)

Ensure compliance with mobile storefronts

- Static code analysis
- Supply-chain vulnerability scans

Ultimately, ensure user satisfaction through security


OWASP
®

OWASP Mobile Top 10

Refers to the top 10 risks for mobile applications

- M1: Improper Credential Usage
- M2: Inadequate Supply Chain Security
- M3: Insecure Authentication/Authorization
- M4: Insufficient Input/Output Validation
- M5: Insecure Communication
- M6: Inadequate Privacy Controls
- M7: Insufficient Binary Protections
- M8: Security Misconfiguration
- M9: Insecure Data Storage
- M10: Insufficient Cryptography


MOBSF

MobSF

MobSF is a cost-effective, OWASP-compliant tool to scan mobile applications for threats.

Open-Source

OWASP-Compliant

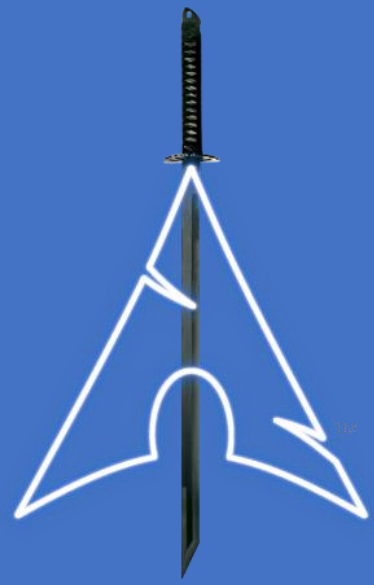
Community-and commercially-supported

Easy to integrate to CI/CD

Detailed reporting

MobSF + Linux

Pentoo



Black Arch

MobSF + Docker

mobsfscan.yml

```
docker pull opensecurity/mobile-security-framework-mobsf:latest
```

Demo:
Static Analysis

mobsfscan

Static analysis tool for automating scanning

Supports popular CI/CD platforms

- Github Actions
- Gitlab CI/CD
- Travis CI
- Azure DevOps

Supports native programming languages

- Java
- Kotlin
- Swift
- Objective-C

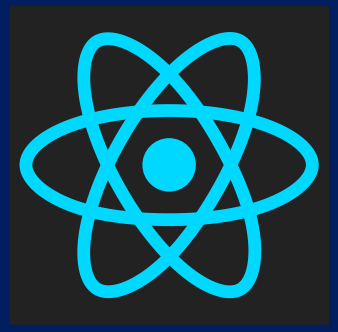
mobsfscan

```yaml
- script: >
    mobsfscan --no-fail --html
      -o $(System.ArtifactsDirectory)/report.html
      $(System.ArtifactsDirectory)/InsecureBankv2/
  displayName: 'Run MobSF Scan'

- publish: $(System.ArtifactsDirectory)/report.html
  artifact: MobSFScanReport
  displayName: 'Publish MobSF Scan Report'
```

azure-pipelines.yml

```yaml
trigger:
- none

pr:
- none

pool:
  name: Self-hosted

jobs:
- job: Publish
  displayName: 'Publish Job'
  pool:
    name: Self-hosted
  steps:
  - task: PublishBuildArtifacts@1
    inputs:
      pathtoPublish: 'InsecureBankv2/app/src/'
      artifactName: 'InsecureBankv2'

  - task: DownloadBuildArtifacts@0
    inputs:
      buildType: 'current'
      downloadType: 'single'
      artifactName: 'InsecureBankv2'
      downloadPath: '$(System.ArtifactsDirectory)'

  - script: >
      mobsfscan --no-fail --html
        -o $(System.ArtifactsDirectory)/report.html
        $(System.ArtifactsDirectory)/InsecureBankv2/
    displayName: 'Run MobSF Scan'

  - publish: $(System.ArtifactsDirectory)/report.html
    artifact: MobSFScanReport
    displayName: 'Publish MobSF Scan Report'
```

What about cross-platform frameworks?



Scanning compiled binaries

- Build app
- Copy artifacts
- Pull *mobsf* docker image
- Run image
- Upload artifact
- Execute scan
- Pull report

mobsf-scan.yml

```yaml
- script: docker pull opensecurity/mobile-security-framework-mobsf:latest
  displayName: 'Pull MobSF Docker Image'
  condition: succeeded()

- script: >
    docker run -d -it --rm -e MOBSF_API_KEY='${{parameters.mobSfApiKey}}'
    -e DATA_UPLOAD_MAX_MEMORY_SIZE=209715200
    -p 8000:8000 opensecurity/mobile-security-framework-mobsf:latest
  displayName: 'Run MobSF Docker Image'
  condition: succeeded()

- script: >
    curl -X POST
    http://127.0.0.1:8000/api/v1/upload
    -F "file=@$file;type=application/octet-stream"
    -H "Authorization: ${{parameters.mobSfApiKey}}"
  displayName: 'Upload to MobSF'
  condition: succeeded()

- script: >
    curl -X POST
    http://127.0.0.1:8000/api/v1/scan
    -H "Authorization: ${{parameters.mobSfApiKey}}"
    --data "scan_type=$($scanBody.scan_type)&hash=$($scanBody.hash)"
  displayName: 'Initiate scan'
  condition: succeeded()

- script: >
    curl -X POST
    http://127.0.0.1:8000/api/v1/download_pdf
    -H "Authorization: ${{parameters.mobSfApiKey}}"
    --data "hash=$($scanBody.hash)"
  displayName: 'Download PDF report'
  condition: succeeded()
```

Demo:
Automated Scanning

Source code:

Merci !
Thank you!

Feedback: