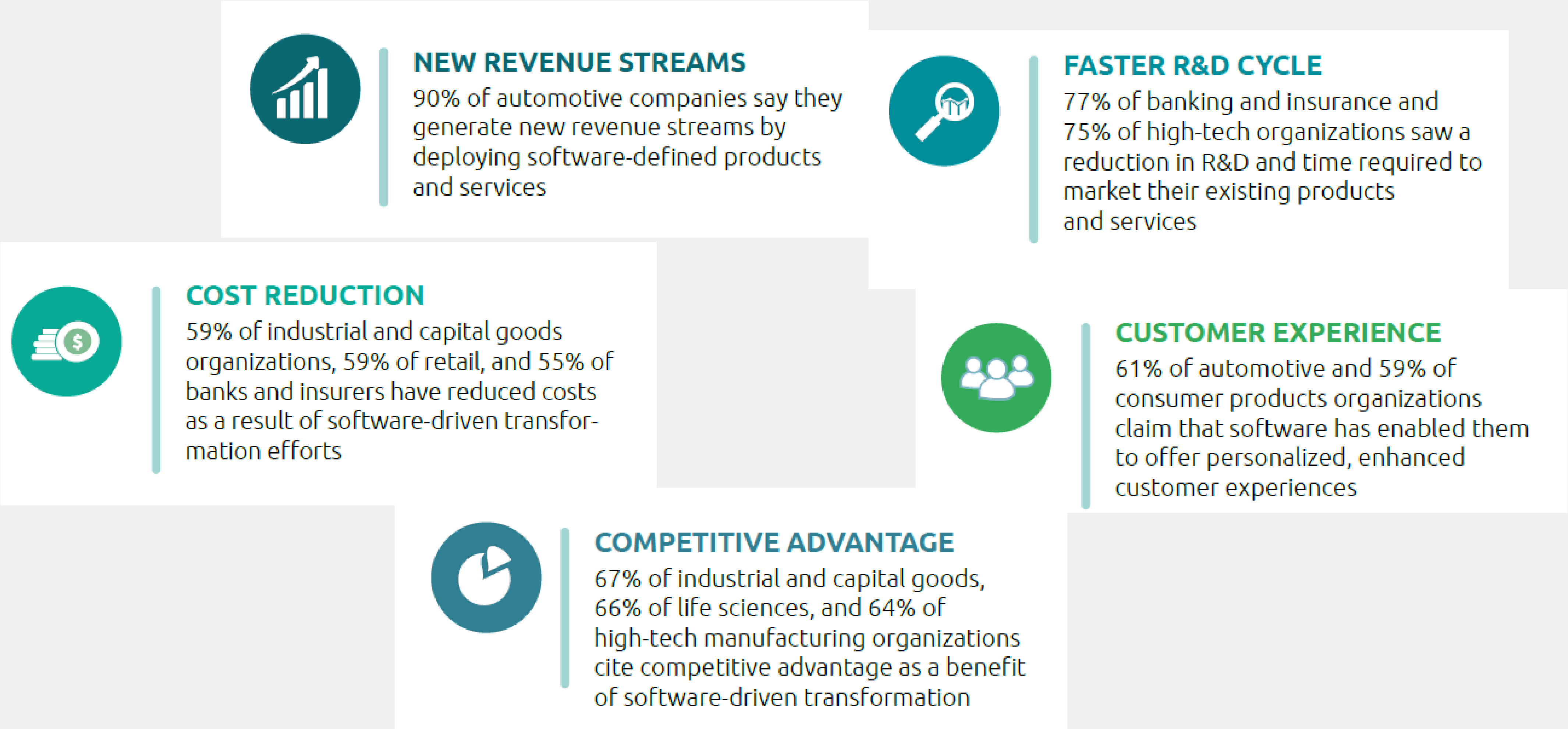
FORTINET

# Web Application Security: From Development to Production

Sander Ruiter
Telco / MSSP System Engineer
Fortinet

# Software Powers Organizations…



**NEW REVENUE STREAMS**

90% of automotive companies say they generate new revenue streams by deploying software-defined products and services

**FASTER R&D CYCLE**

77% of banking and insurance and 75% of high-tech organizations saw a reduction in R&D and time required to market their existing products and services

**COST REDUCTION**

59% of industrial and capital goods organizations, 59% of retail, and 55% of banks and insurers have reduced costs as a result of software-driven transformation efforts

**CUSTOMER EXPERIENCE**

61% of automotive and 59% of consumer products organizations claim that software has enabled them to offer personalized, enhanced customer experiences

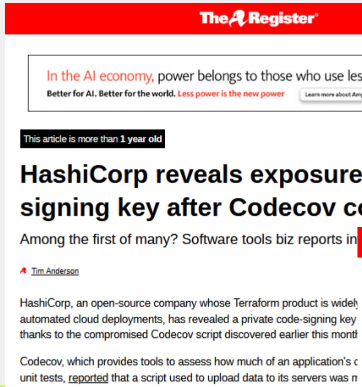
**COMPETITIVE ADVANTAGE**

67% of industrial and capital goods, 66% of life sciences, and 64% of high-tech manufacturing organizations cite competitive advantage as a benefit of software-driven transformation

https://www.capgemini.com/insights/research-library/softwarization-research/

# What can go wrong in Software world??

Reputation Risks, PSIRT issues, PII breaches etc.…



The Register

This article is more than **1 year old**

## HashiCorp reveals exposure of private code-signing key after Codecov compromise

Among the first of many? Software tools biz reports in

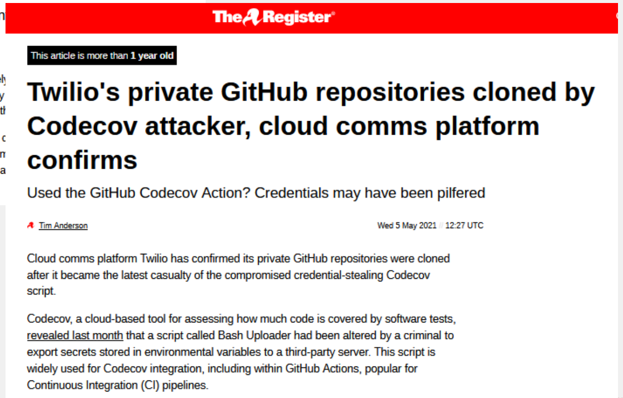
Tim Anderson

HashiCorp, an open-source company whose Terraform product is widel
automated cloud deployments, has revealed a private code-signing key
thanks to the compromised Codecov script discovered earlier this montl

Codecov, which provides tools to assess how much of an application's c
unit tests, reported that a script used to upload data to its servers was m
credentials to an attacker's server. The company said it had "not been a
conclusively who carried out the event."

HashiCorp

twilio



The Register

This article is more than **1 year old**

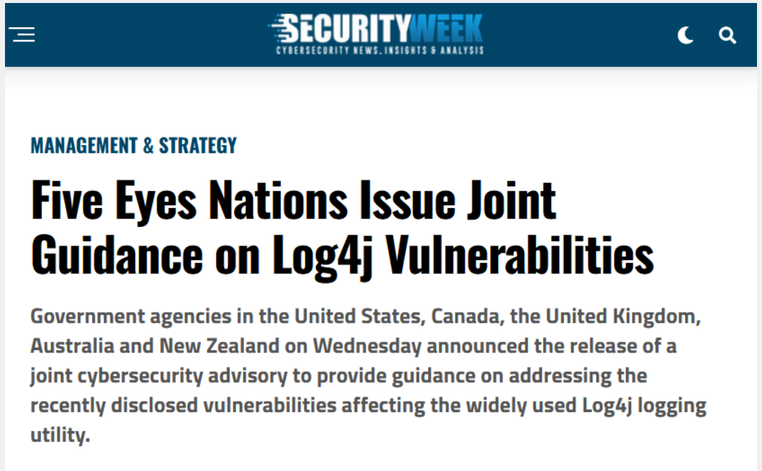
## Twilio's private GitHub repositories cloned by Codecov attacker, cloud comms platform confirms

Used the GitHub Codecov Action? Credentials may have been pilfered

Tim Anderson Wed 5 May 2021 // 12:27 UTC

Cloud comms platform Twilio has confirmed its private GitHub repositories were cloned after it became the latest casualty of the compromised credential-stealing Codecov script.

Codecov, a cloud-based tool for assessing how much code is covered by software tests, revealed last month that a script called Bash Uploader had been altered by a criminal to export secrets stored in environmental variables to a third-party server. This script is widely used for Codecov integration, including within GitHub Actions, popular for Continuous Integration (CI) pipelines.

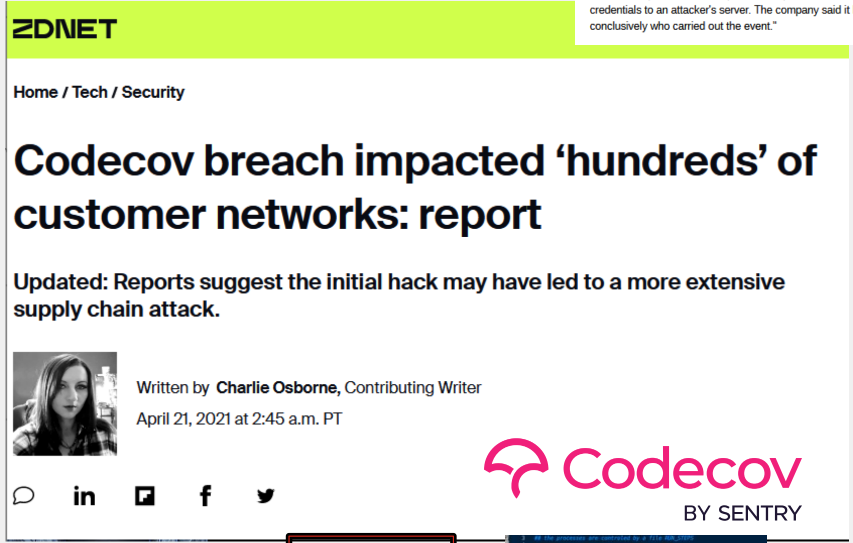


SECURITYWEEK
CYBERSECURITY NEWS, INSIGHTS & ANALYSIS

MANAGEMENT & STRATEGY

## Five Eyes Nations Issue Joint Guidance on Log4j Vulnerabilities

Government agencies in the United States, Canada, the United Kingdom, Australia and New Zealand on Wednesday announced the release of a joint cybersecurity advisory to provide guidance on addressing the recently disclosed vulnerabilities affecting the widely used Log4j logging utility.



ZDNET

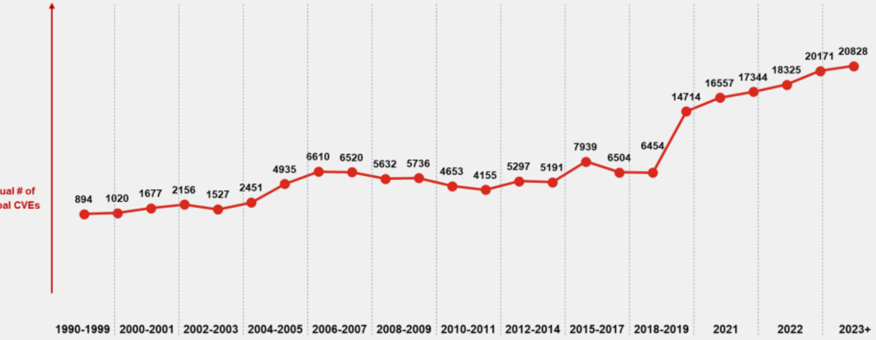
Home / Tech / Security

## Codecov breach impacted 'hundreds' of customer networks: report

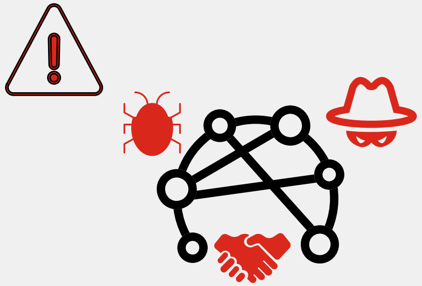
**Updated: Reports suggest the initial hack may have led to a more extensive supply chain attack.**

Written by **Charlie Osborne,** Contributing Writer
April 21, 2021 at 2:45 a.m. PT

Codecov
BY SENTRY



Software Vulnerabilities

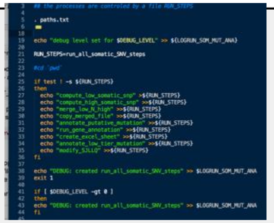


Software Supply Chain Vulnerabilities & Attacks

[https://www.csoonline.com/article/570743/6-most-common-types-of-software-supply-chain-attacks-explained.html](https://www.csoonline.com/article/570743/6-most-common-types-of-software-supply-chain-attacks-explained.html)

USERNAME
✱✱✱✱✱

docker



```
## the processes are controled by a file RUN_STEPS

. paths.txt


echo "debug level set for $DEBUG_LEVEL" >> ${LOGRUN_SOM_MUT_ANA}

RUN_STEPS=run_all_somatic_SNV_steps

#cd `pwd`

if test ! -s ${RUN_STEPS}
then
  echo "compute_low_somatic_snp" >${RUN_STEPS}
  echo "compute_high_somatic_snp" >>${RUN_STEPS}
  echo "merge_low_N_high" >>${RUN_STEPS}
  echo "copy_merged_file" >>${RUN_STEPS}
  echo "annotate_putative_mutation" >>${RUN_STEPS}
  echo "run_gene_annotation" >>${RUN_STEPS}
  echo "create_excel_sheet" >>${RUN_STEPS}
  echo "annotate_low_tier_mutation" >>${RUN_STEPS}
  echo "modify_SJLLQ" >>${RUN_STEPS}
fi

echo "DEBUG: created run_all_somatic_SNV_steps" >> $LOGRUN_SOM_MUT_ANA
exit 1

if [ $DEBUG_LEVEL -gt 0 ]
then
echo "DEBUG: created run_all_somatic_SNV_steps" >> $LOGRUN_SOM_MUT_ANA
fi
```

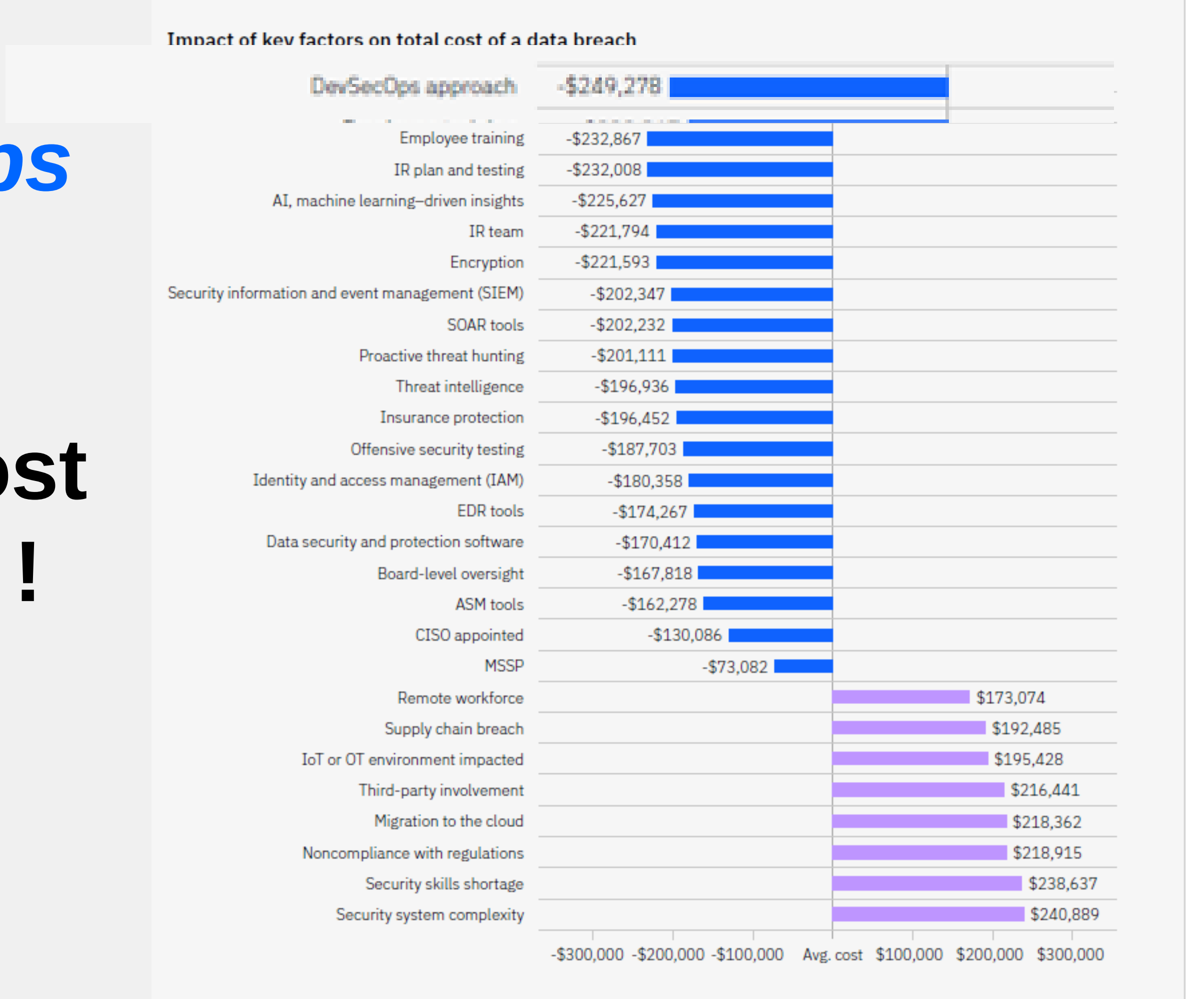
# Best way to reduce the cost of a breach?

Where to invest for maximum impact

***DevSecOps***

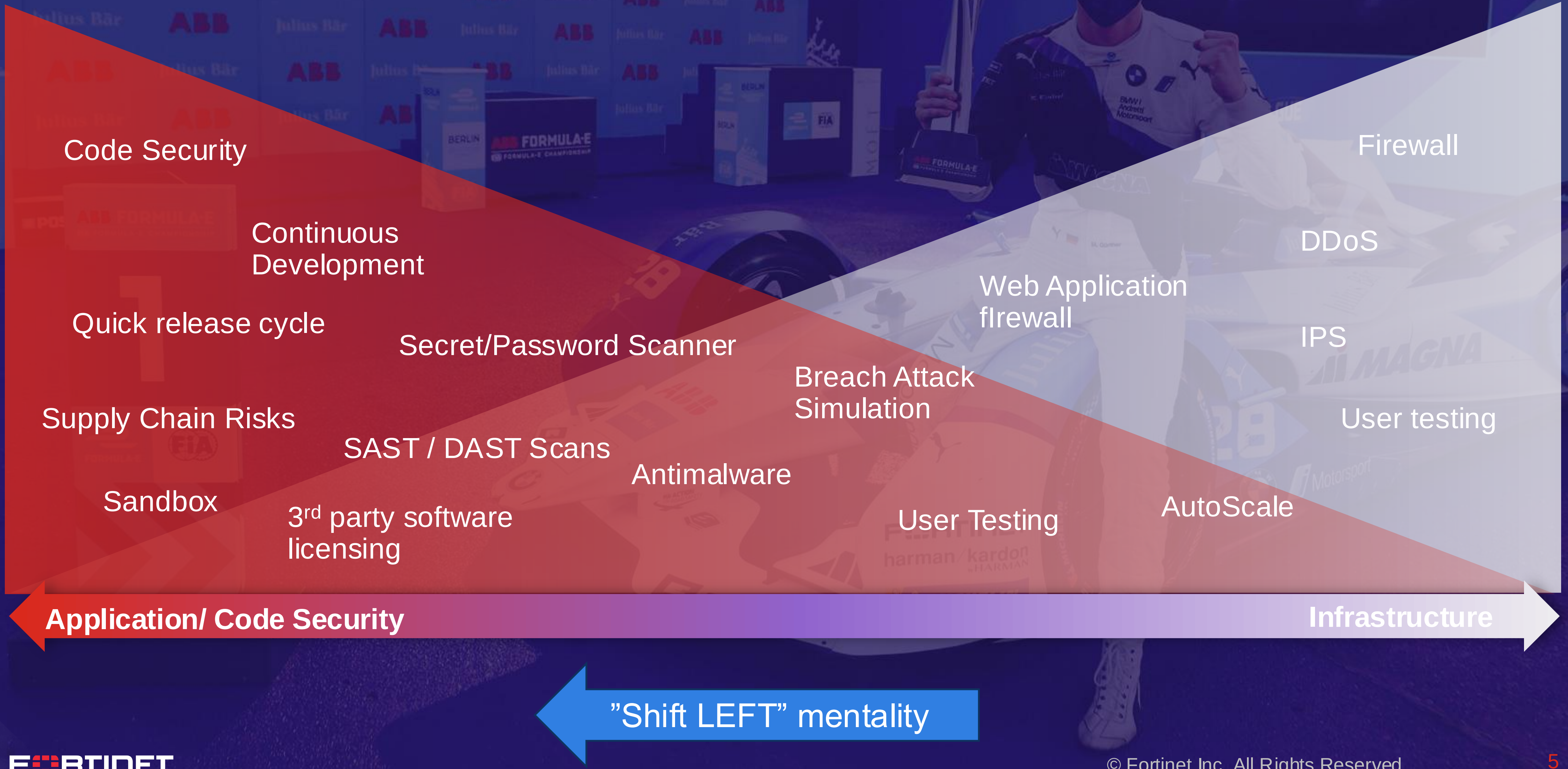
**TOP**

**Breach Cost Mitigator !**


Impact of key factors on total cost of a data breach

IBM Cost of a Data Breach  Report 2023

# Redefining Application Security



Code Security

Continuous Development

Quick release cycle

Secret/Password Scanner

Supply Chain Risks

SAST / DAST Scans

Sandbox

3rd party software licensing

Antimalware

Breach Attack Simulation

Web Application fIrewall

User Testing

AutoScale

Firewall

DDoS

IPS

User testing

**Application/ Code Security**

**Infrastructure**

”Shift LEFT” mentality

# Driving modern software development practices

Agile DevOps practice with shorter release cycles, continuous, fast, automated process

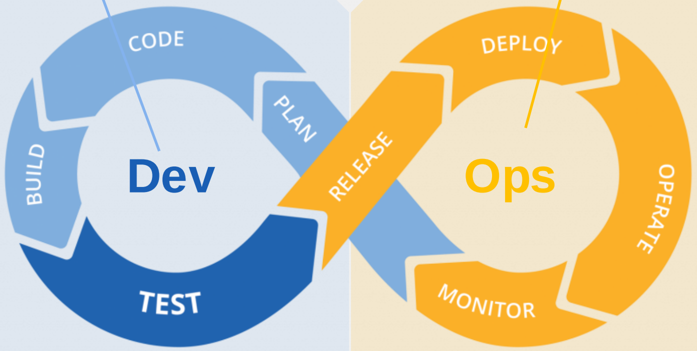
**Development**

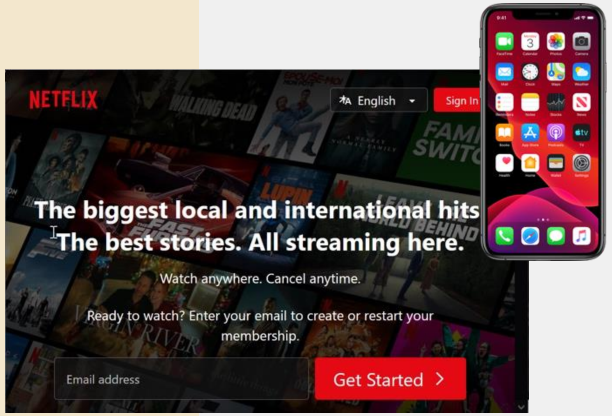
= Write
= Build

**Operations**

= Deploy, Run
= Production

```
})
ws.on("message", m => {
  let a = m.split(" ")
  switch(a[0]){
    case "connect":
      if(a[1]){
        if(clients.has(a[1])){
          ws.send("connected");
          ws.id = a[1];
        }else{
          ws.id = a[1]
          clients.set(a[1], {client: {position:
          ws.send("connected")
        }
      }else{
        let id = Math.random().toString()
        ws.id = id;
        clients.set(id, {client: {position:
```





NETFLIX

English

Sign In

The biggest local and international hits. The best stories. All streaming here.

Watch anywhere. Cancel anytime.

Ready to watch? Enter your email to create or restart your membership.

Email address

Get Started

CI

CD

***Speed >>***

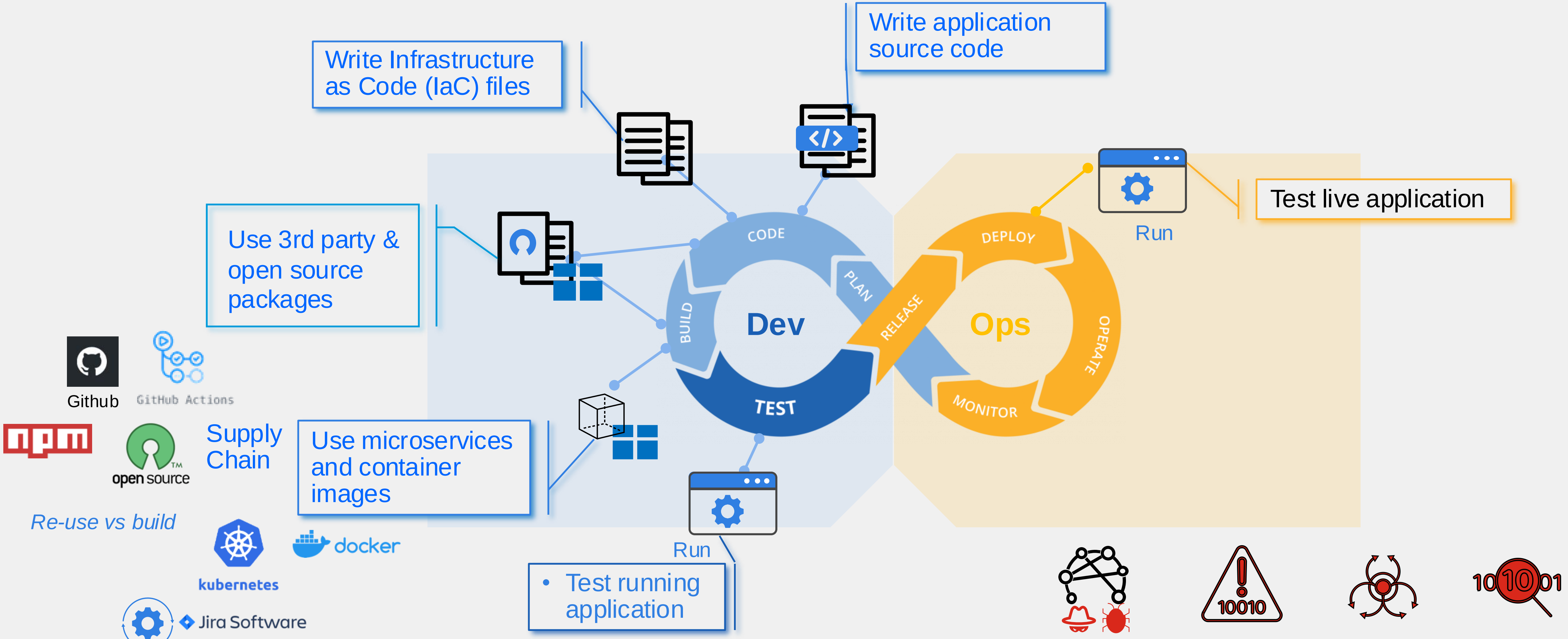
<< Continuous Integration << **CI / CD** >> Continuous Deployment >>

70% of DevOps teams release code continuously, once a day, or every few days

The Gitlab 2022 Global DevSecOps Survey

# How are software applications built?

## Modern software development lifecycle (SDLC)



Write Infrastructure as Code (IaC) files

Write application source code

Use 3rd party & open source packages

Test live application

Supply Chain

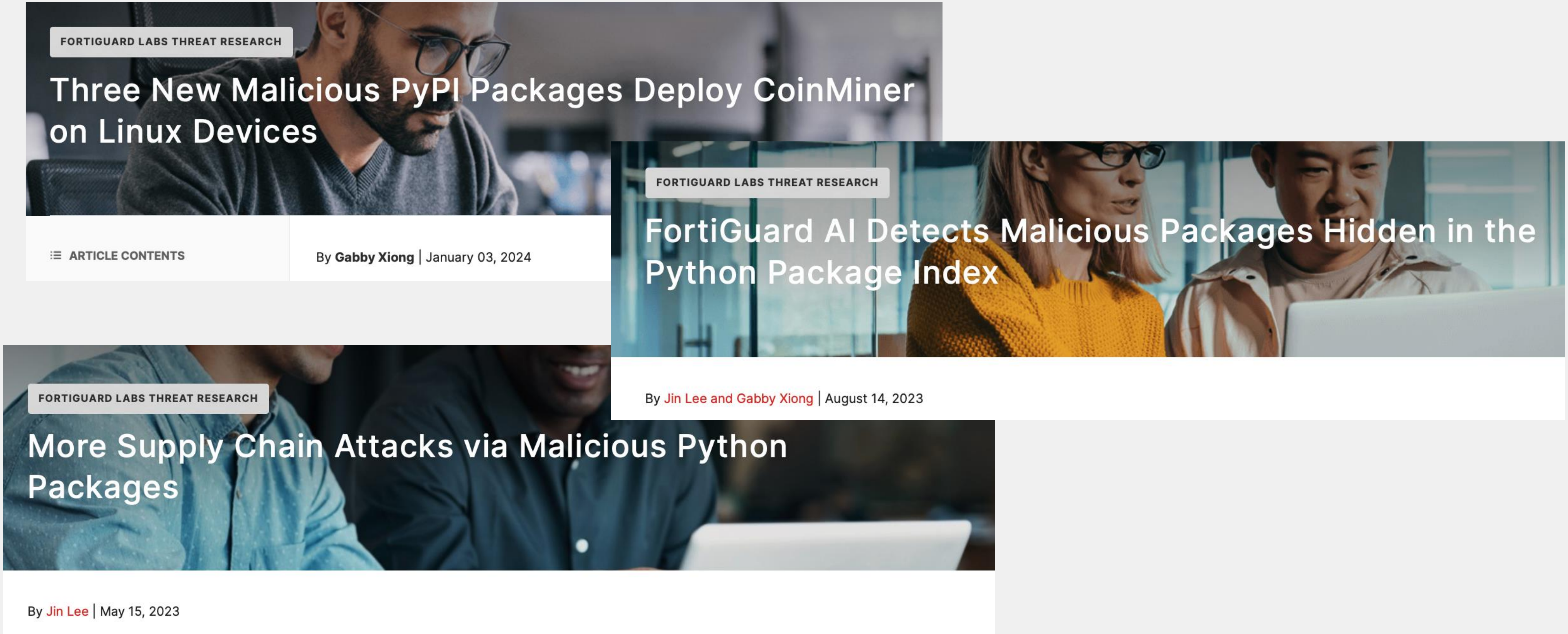
Use microservices and container images

*Re-use vs build*

- Test running application

Multiple Risk Points with Supply Chain, Code Security, malware, malicous packages

# FortiGuard Research in Supply Chain Risk



FORTIGUARD LABS THREAT RESEARCH

Three New Malicious PyPI Packages Deploy CoinMiner on Linux Devices

ARTICLE CONTENTS

By Gabby Xiong | January 03, 2024

FORTIGUARD LABS THREAT RESEARCH

FortiGuard AI Detects Malicious Packages Hidden in the Python Package Index

By Jin Lee and Gabby Xiong | August 14, 2023

FORTIGUARD LABS THREAT RESEARCH

More Supply Chain Attacks via Malicious Python Packages

By Jin Lee | May 15, 2023

https://www.fortinet.com/blog/threat-research/malicious-pypi-packages-deploy-coinminer-on-linux-devices

# Who is concerned about Application/Code Security?

Lack of Application Security Expertise

## DevOps engineers

- Build/Compile software
- Control deployment /pipeline management
- UAT/production release
- Automates security in CI/CD

## CISO

- Application Security / Publicity / Brand awareness

## Application Owners

- Agile & Secure Development
- Less PSIRT issues

## Developers

- Balance of bug fixes and new features
- Different level of expertise re secure coding practices

# Shift-left from Web App Security to Code Security

# DevSecOps Security

Continuous Integration and Continuous Deployment (CI/CD)

Source Code Scanning (SAST)

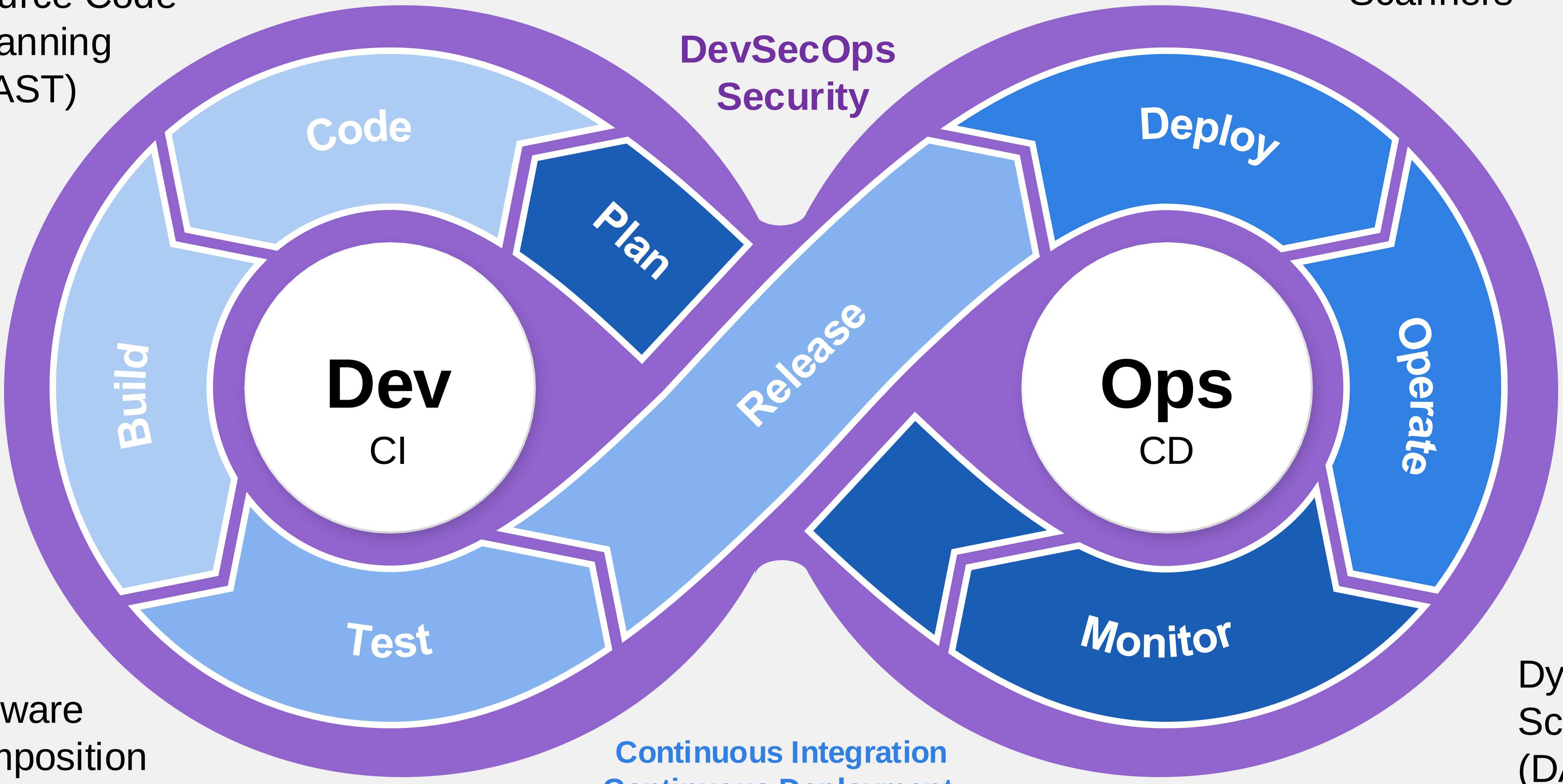
Secrets Scanners

Infrastructure as Code Scanning

Integration Into Build Tools

Software Composition Analysis

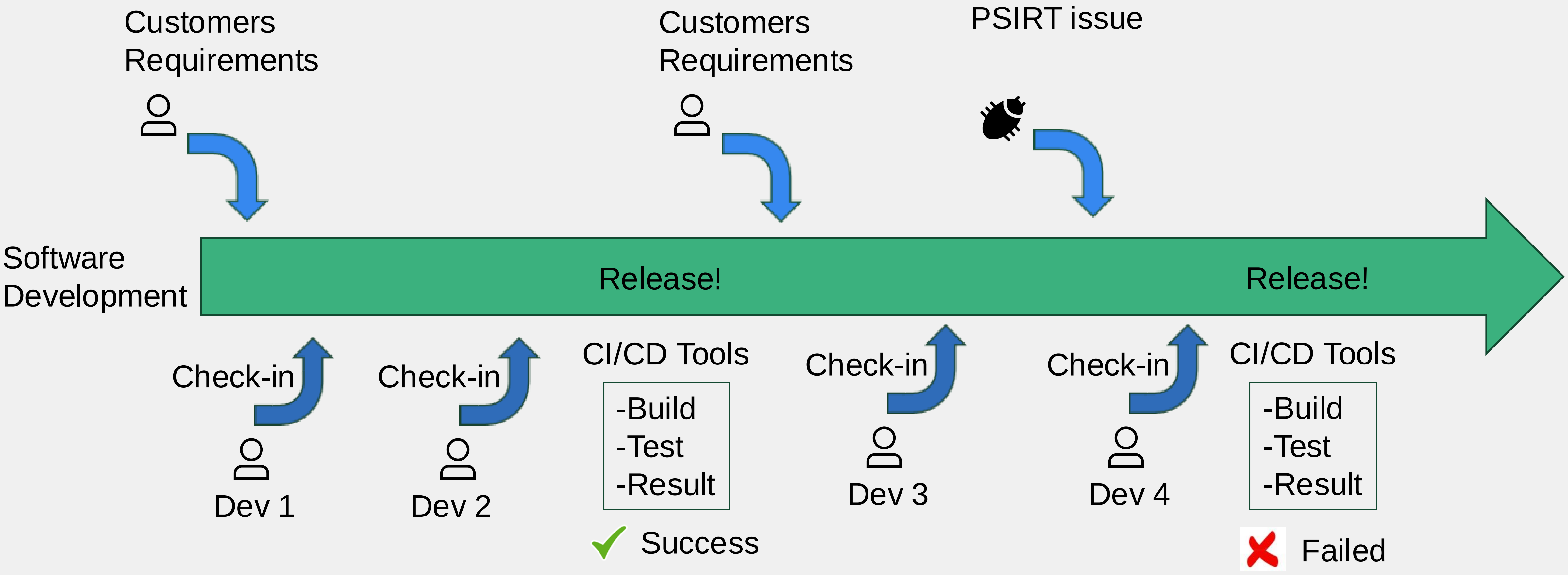
Dynamic App Scans (DAST)



Continuous Integration
Continuous Deployment
(CI/CD)

# Software Deployment Life Cycle

SDLC – Simple High Level View of Software Cycle



Customers Requirements

Customers Requirements

PSIRT issue

Software Development

Release!

Release!

Check-in

Dev 1

Check-in

Dev 2

CI/CD Tools

-Build
-Test
-Result

✓ Success

Check-in

Dev 3

Check-in

Dev 4

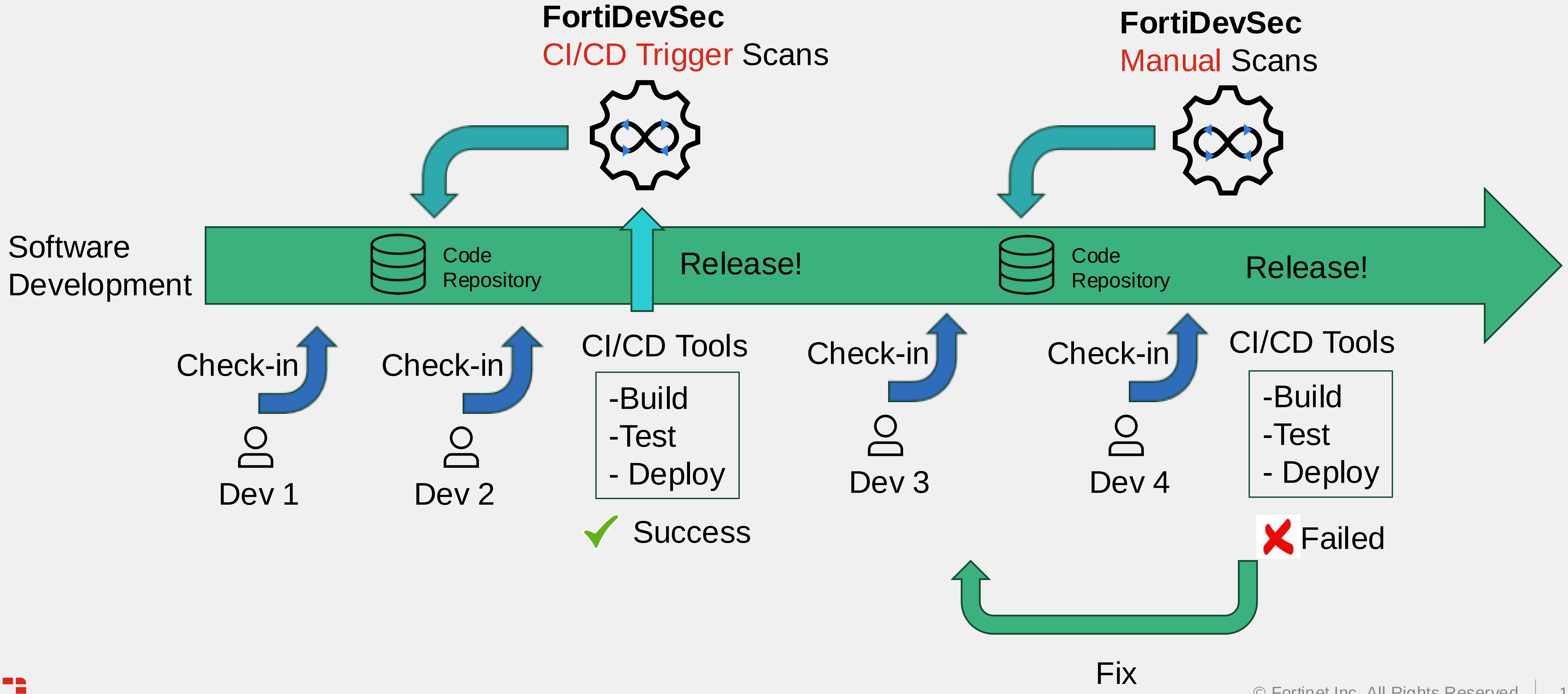
CI/CD Tools

-Build
-Test
-Result

✗ Failed

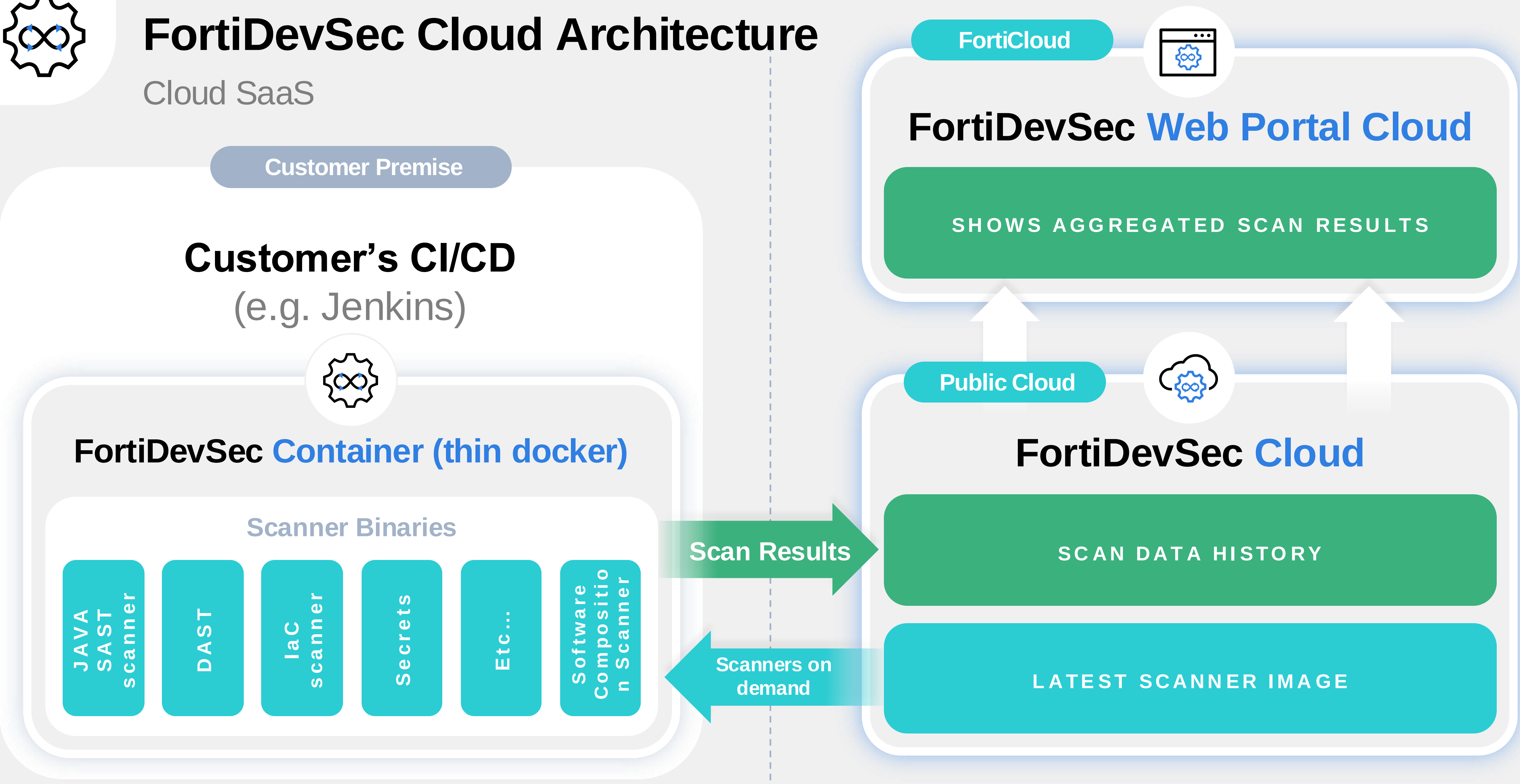
# SECURE - SDLC

Building Security Into Software Development Cycle



# FortiDevSec Cloud Architecture

Cloud SaaS



Customer Premise

Customer's CI/CD
(e.g. Jenkins)

FortiDevSec Container (thin docker)

Scanner Binaries

JAVA SAST scanner

DAST

IaC scanner

Secrets

Etc…

Software Composition Scanner

Scan Results

Scanners on demand

FortiCloud

FortiDevSec Web Portal Cloud

SHOWS AGGREGATED SCAN RESULTS

Public Cloud

FortiDevSec Cloud

SCAN DATA HISTORY

LATEST SCANNER IMAGE

# FortiDevSec
# Types of Scans Available

# FortiDevSec - Types of Scans Available

See issues aggregated across multiple types of scanning

## SAST

**Static / source code scanning (SAST) –**
issues in application source code

Supports *Shell, Java, Ruby on Rails, Python, Golang, PHP, JavaScript/NodeJS, C, C++* and *C# .Net.*

## SCA/OSS

**SCA/OSS scanning –**
issues in third party and open source libraries e.g. log4j

Identify Outbreak and Supply Chain Attacks

## Secrets

**Secrets –**
scans for open password text

## DAST

**Dynamic scanning (DAST) –**
simulates exploits using application's front end url, using FortiDAST product add-on

## Containers

**Scanning Containers**
that are built in the pipeline

## Infrastructure as Code

**Infrastructure as Script security scanning –**
scanning IaC scripts like terraform, etc.

Supports Terraform, Cloud Formation, Docker and Kubernetes

# FortiDevSec Secrets Scanner

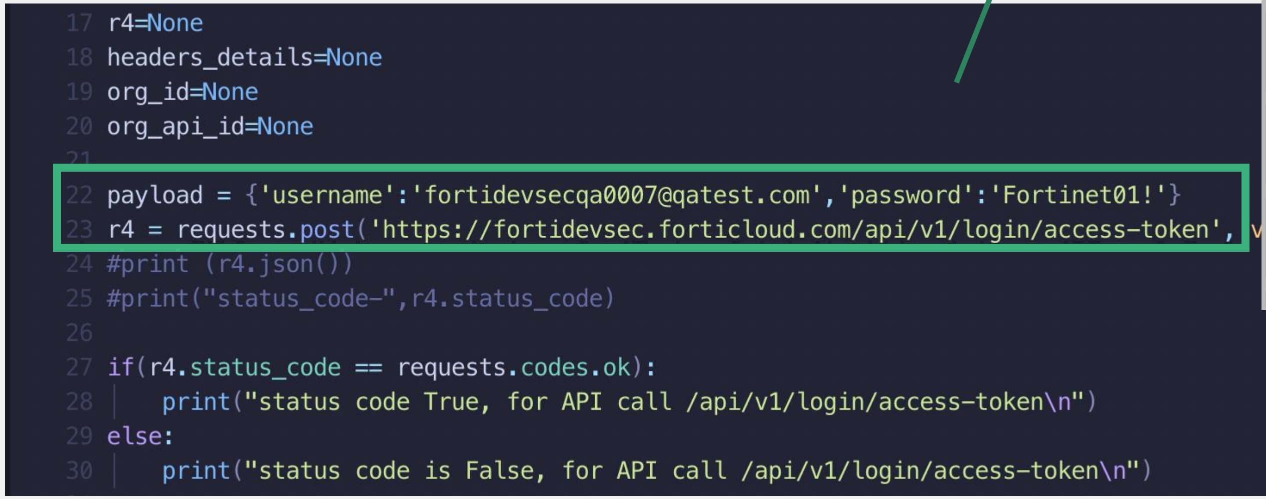
## *Purpose*

To identify hardcoded passwords, PII information in part of source code, code build history. (committed lines of code)

### *Sample Result*

Cleartext secrets discovered in code

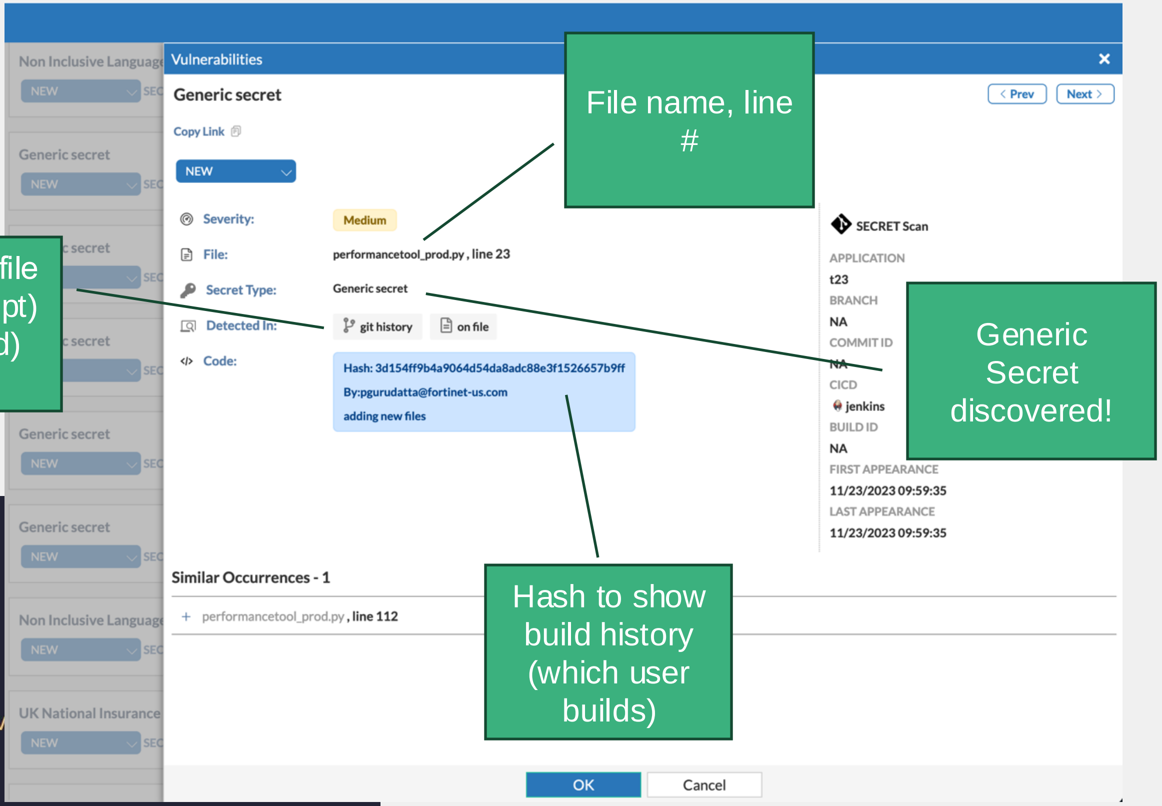
*File performancetool_prod.py* line 23



```
17 r4=None
18 headers_details=None
19 org_id=None
20 org_api_id=None
21
22 payload = {'username':'fortidevsecqa0007@qatest.com','password':'Fortinet01!'}
23 r4 = requests.post('https://fortidevsec.forticloud.com/api/v1/login/access-token', v
24 #print (r4.json())
25 #print("status_code-",r4.status_code)
26
27 if(r4.status_code == requests.codes.ok):
28     print("status code True, for API call /api/v1/login/access-token\n")
29 else:
30     print("status code is False, for API call /api/v1/login/access-token\n")
```

Present in ”file (python script) & Git (build) history

*Figure – FortiDevSec Secrets Scan result*



Generic secret

Severity: Medium

File: performancetool_prod.py , line 23

Secret Type: Generic secret

Detected In: git history | on file

Code:
Hash: 3d154ff9b4a9064d54da8adc88e3f1526657b9ff
By:pgurudatta@fortinet-us.com
adding new files

SECRET Scan

APPLICATION
t23

BRANCH
NA

COMMIT ID
NA

CICD
jenkins

BUILD ID
NA

FIRST APPEARANCE
11/23/2023 09:59:35

LAST APPEARANCE
11/23/2023 09:59:35

Similar Occurrences - 1

+ performancetool_prod.py , line 112

File name, line #

Generic Secret discovered!

Hash to show build history (which user builds)

# FortiDevSec SCA Scanner

Software Composition Analysis

*FortiDevSec shows SBOM (software bills of material used)*

## *Purpose*

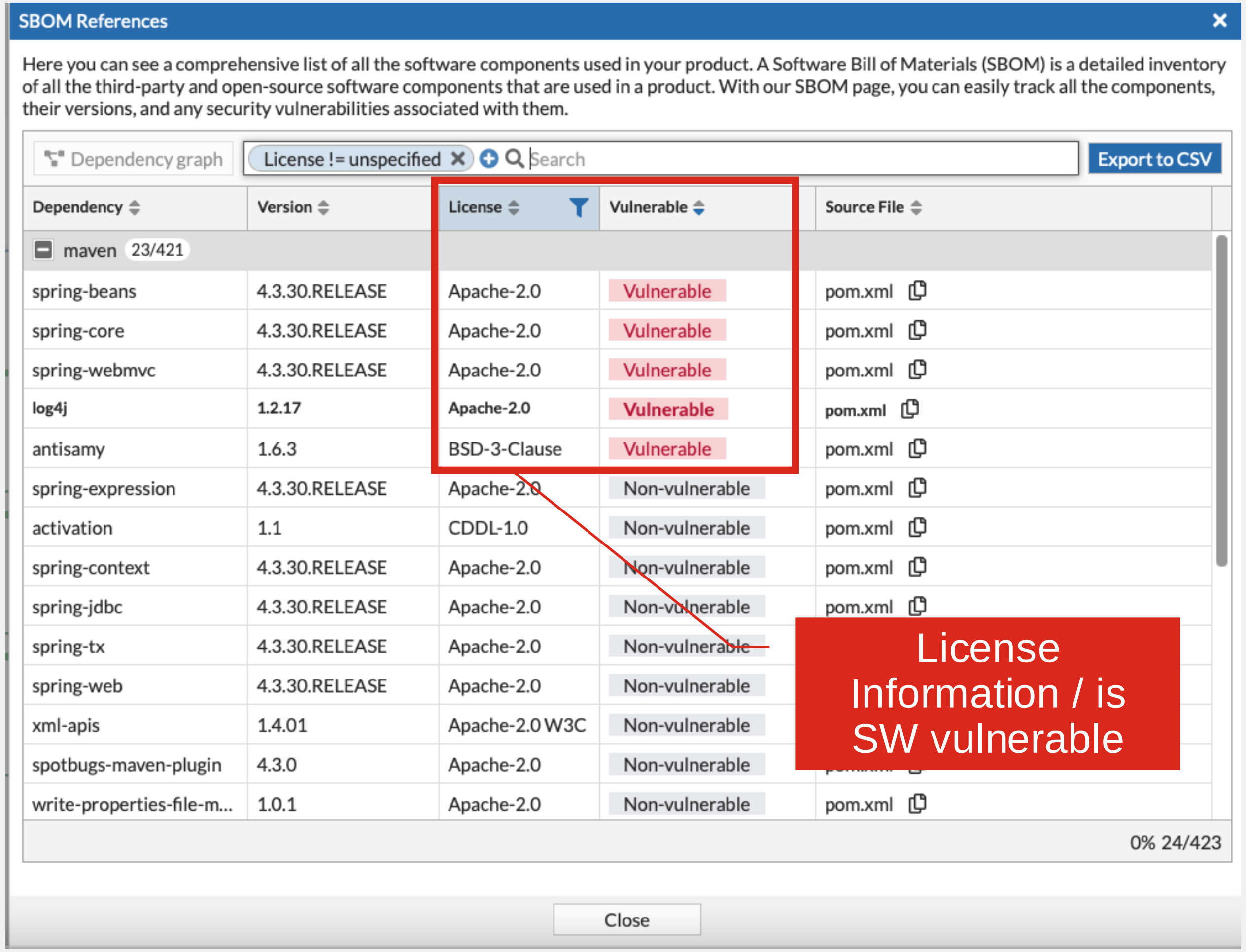
Scans for vulnerabilities in the **open-source libraries/components** used by the application. The programming languages supported by the SCA scanner are *Java, Javascript, Ruby, Python, Golang, C# .Net and PHP*.

### *Sample Result*

Identifies all 3rd party libraries, one of vulnerable Apache version

Could be Intellectual Property violation that can lead into legal lawsuits!

Software BOM reference



SBOM References

Here you can see a comprehensive list of all the software components used in your product. A Software Bill of Materials (SBOM) is a detailed inventory of all the third-party and open-source software components that are used in a product. With our SBOM page, you can easily track all the components, their versions, and any security vulnerabilities associated with them.

| Dependency | Version | License | Vulnerable | Source File |
|---|---|---|---|---|
| maven 23/421 | | | | |
| spring-beans | 4.3.30.RELEASE | Apache-2.0 | Vulnerable | pom.xml |
| spring-core | 4.3.30.RELEASE | Apache-2.0 | Vulnerable | pom.xml |
| spring-webmvc | 4.3.30.RELEASE | Apache-2.0 | Vulnerable | pom.xml |
| log4j | 1.2.17 | Apache-2.0 | Vulnerable | pom.xml |
| antisamy | 1.6.3 | BSD-3-Clause | Vulnerable | pom.xml |
| spring-expression | 4.3.30.RELEASE | Apache-2.0 | Non-vulnerable | pom.xml |
| activation | 1.1 | CDDL-1.0 | Non-vulnerable | pom.xml |
| spring-context | 4.3.30.RELEASE | Apache-2.0 | Non-vulnerable | pom.xml |
| spring-jdbc | 4.3.30.RELEASE | Apache-2.0 | Non-vulnerable | pom.xml |
| spring-tx | 4.3.30.RELEASE | Apache-2.0 | Non-vulnerable | |
| spring-web | 4.3.30.RELEASE | Apache-2.0 | Non-vulnerable | |
| xml-apis | 1.4.01 | Apache-2.0 W3C | Non-vulnerable | |
| spotbugs-maven-plugin | 4.3.0 | Apache-2.0 | Non-vulnerable | |
| write-properties-file-m... | 1.0.1 | Apache-2.0 | Non-vulnerable | pom.xml |

0% 24/423

License Information / is SW vulnerable

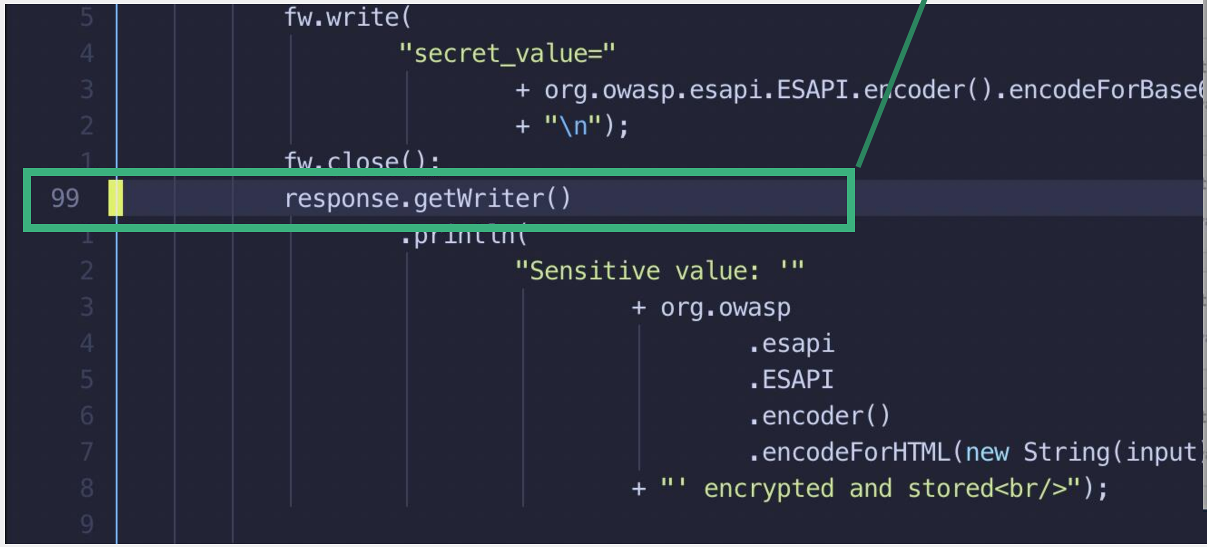
# FortiDevSec SAST Scan for Java Example

Static Application Security Testing

## *Purpose*

To identify vulnerabilities in code, also known as “white-box” testing, usually done before code is compiled. Support multiple languages.

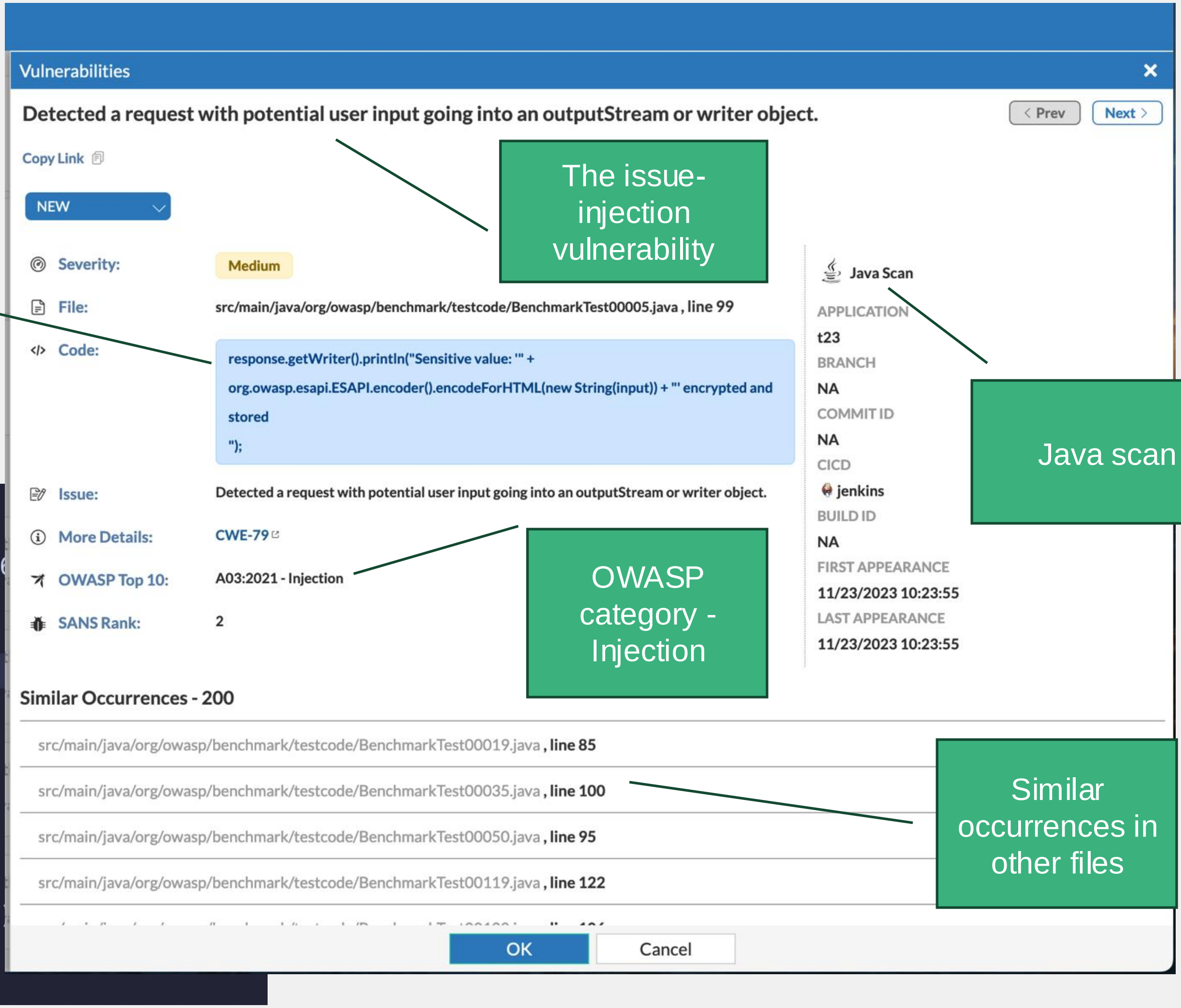
*File BenchmarkTestxxx.java* line 99



```
            fw.write(
                    "secret_value="
                            + org.owasp.esapi.ESAPI.encoder().encodeForBase
                            + "\n");
            fw.close();
99          response.getWriter()
                    .println(
                            "Sensitive value: '"
                                    + org.owasp
                                            .esapi
                                            .ESAPI
                                            .encoder()
                                            .encodeForHTML(new String(input
                                    + "' encrypted and stored<br/>");
```

Display Code ‘snippet”

*Figure – FortiDevSec SAST Scan result*



Vulnerabilities ✕

Detected a request with potential user input going into an outputStream or writer object.

Severity: Medium

File: src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00005.java , line 99

Code:

```
response.getWriter().println("Sensitive value: '" +
org.owasp.esapi.ESAPI.encoder().encodeForHTML(new String(input)) + "' encrypted and
stored
");
```

Issue: Detected a request with potential user input going into an outputStream or writer object.

More Details: CWE-79

OWASP Top 10: A03:2021 - Injection

SANS Rank: 2

Java Scan

APPLICATION
t23

BRANCH
NA

COMMIT ID
NA

CICD
jenkins

BUILD ID
NA

FIRST APPEARANCE
11/23/2023 10:23:55

LAST APPEARANCE
11/23/2023 10:23:55

Similar Occurrences - 200

src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00019.java , line 85

src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00035.java , line 100

src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00050.java , line 95

src/main/java/org/owasp/benchmark/testcode/BenchmarkTest00119.java , line 122

The issue- injection vulnerability

Java scan

OWASP category - Injection

Similar occurrences in other files

Other Languages Supported: *Shell, Java, Ruby on Rails, Python, Golang, PHP, JavaScript/NodeJS, C, C++* and *C# .Net*.

# FortiDevSec Container Scanner

## *Purpose*

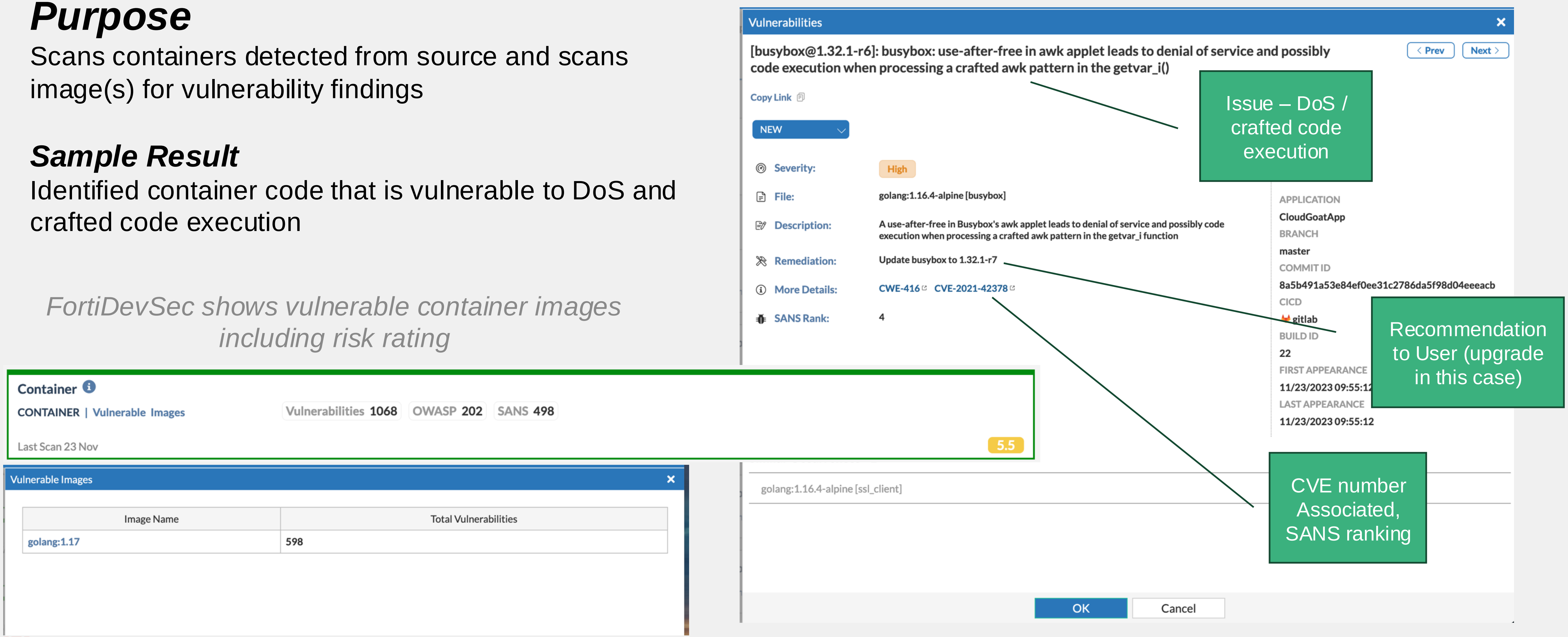
Scans containers detected from source and scans image(s) for vulnerability findings

### *Sample Result*

Identified container code that is vulnerable to DoS and crafted code execution

*FortiDevSec shows vulnerable container images including risk rating*

*Figure – FortiDevSec Container Scan result*



Vulnerabilities ✕

[busybox@1.32.1-r6]: busybox: use-after-free in awk applet leads to denial of service and possibly code execution when processing a crafted awk pattern in the getvar_i()

‹ Prev   Next ›

Copy Link

NEW

Severity: High

File: golang:1.16.4-alpine [busybox]

Description: A use-after-free in Busybox's awk applet leads to denial of service and possibly code execution when processing a crafted awk pattern in the getvar_i function

Remediation: Update busybox to 1.32.1-r7

More Details: CWE-416 CVE-2021-42378

SANS Rank: 4

APPLICATION
CloudGoatApp
BRANCH
master
COMMIT ID
8a5b491a53e84ef0ee31c2786da5f98d04eeeacb
CICD
gitlab
BUILD ID
22
FIRST APPEARANCE
11/23/2023 09:55:12
LAST APPEARANCE
11/23/2023 09:55:12

golang:1.16.4-alpine [ssl_client]

OK   Cancel

Issue – DoS / crafted code execution

Recommendation to User (upgrade in this case)

CVE number Associated, SANS ranking

Container

CONTAINER | Vulnerable Images

Vulnerabilities 1068   OWASP 202   SANS 498

Last Scan 23 Nov

5.5

Vulnerable Images ✕

| Image Name | Total Vulnerabilities |
|---|---|
| golang:1.17 | 598 |

# FortiDevSec IaC Scanner – Terraform example

Infrastructure as Code

## *Purpose*

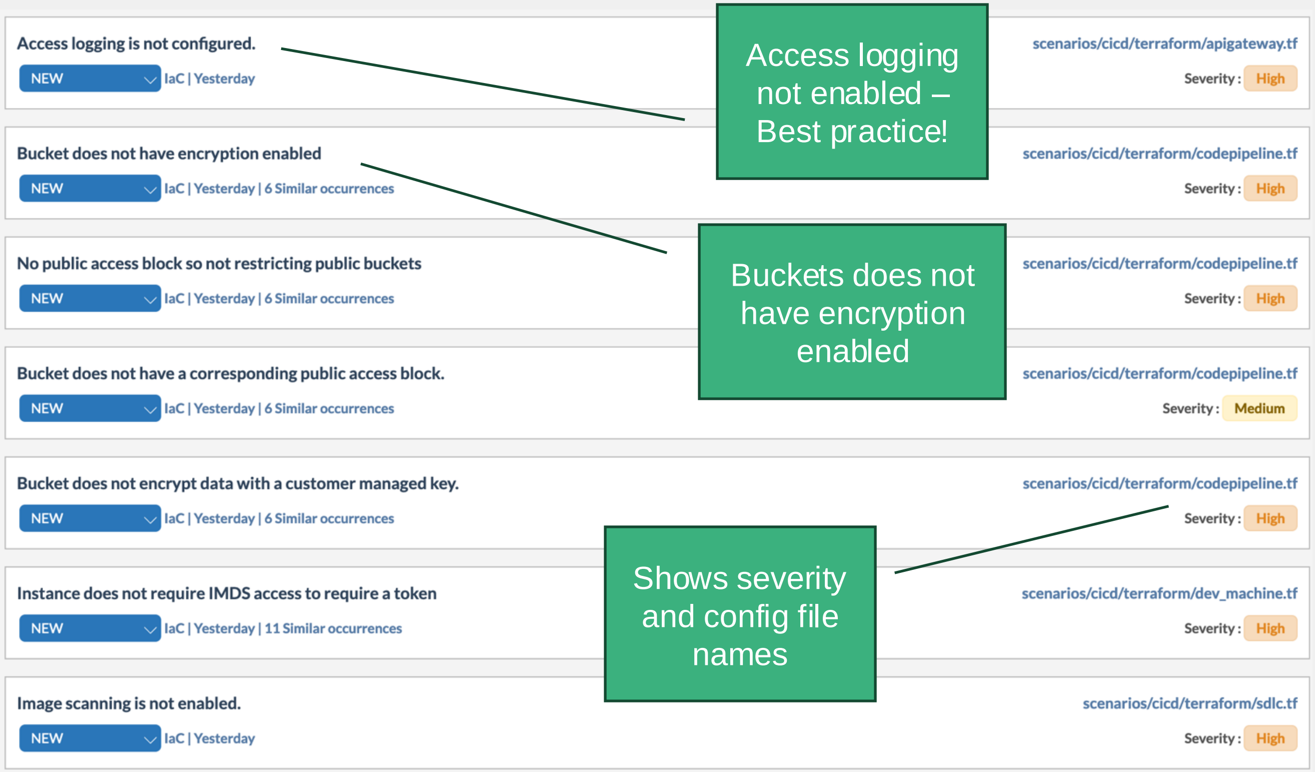
Scans your IaC configuration files from *Terraform, Cloud Formation, Docker* and *Kubernetes* to detect configuration issues.

### *Sample Result*

Identified multiple configuration issues with Terraform configuration file

Other IaC support: Terraform, Cloud Formation, Docker and Kubernetes

*Figure – FortiDevSec IaC (terraform) Scan result*



Access logging is not configured.
NEW | IaC | Yesterday
scenarios/cicd/terraform/apigateway.tf
Severity : High

Bucket does not have encryption enabled
NEW | IaC | Yesterday | 6 Similar occurrences
scenarios/cicd/terraform/codepipeline.tf
Severity : High

No public access block so not restricting public buckets
NEW | IaC | Yesterday | 6 Similar occurrences
scenarios/cicd/terraform/codepipeline.tf
Severity : High

Bucket does not have a corresponding public access block.
NEW | IaC | Yesterday | 6 Similar occurrences
scenarios/cicd/terraform/codepipeline.tf
Severity : Medium

Bucket does not encrypt data with a customer managed key.
NEW | IaC | Yesterday | 6 Similar occurrences
scenarios/cicd/terraform/codepipeline.tf
Severity : High

Instance does not require IMDS access to require a token
NEW | IaC | Yesterday | 11 Similar occurrences
scenarios/cicd/terraform/dev_machine.tf
Severity : High

Image scanning is not enabled.
NEW | IaC | Yesterday
scenarios/cicd/terraform/sdlc.tf
Severity : High

Access logging not enabled – Best practice!

Buckets does not have encryption enabled

Shows severity and config file names

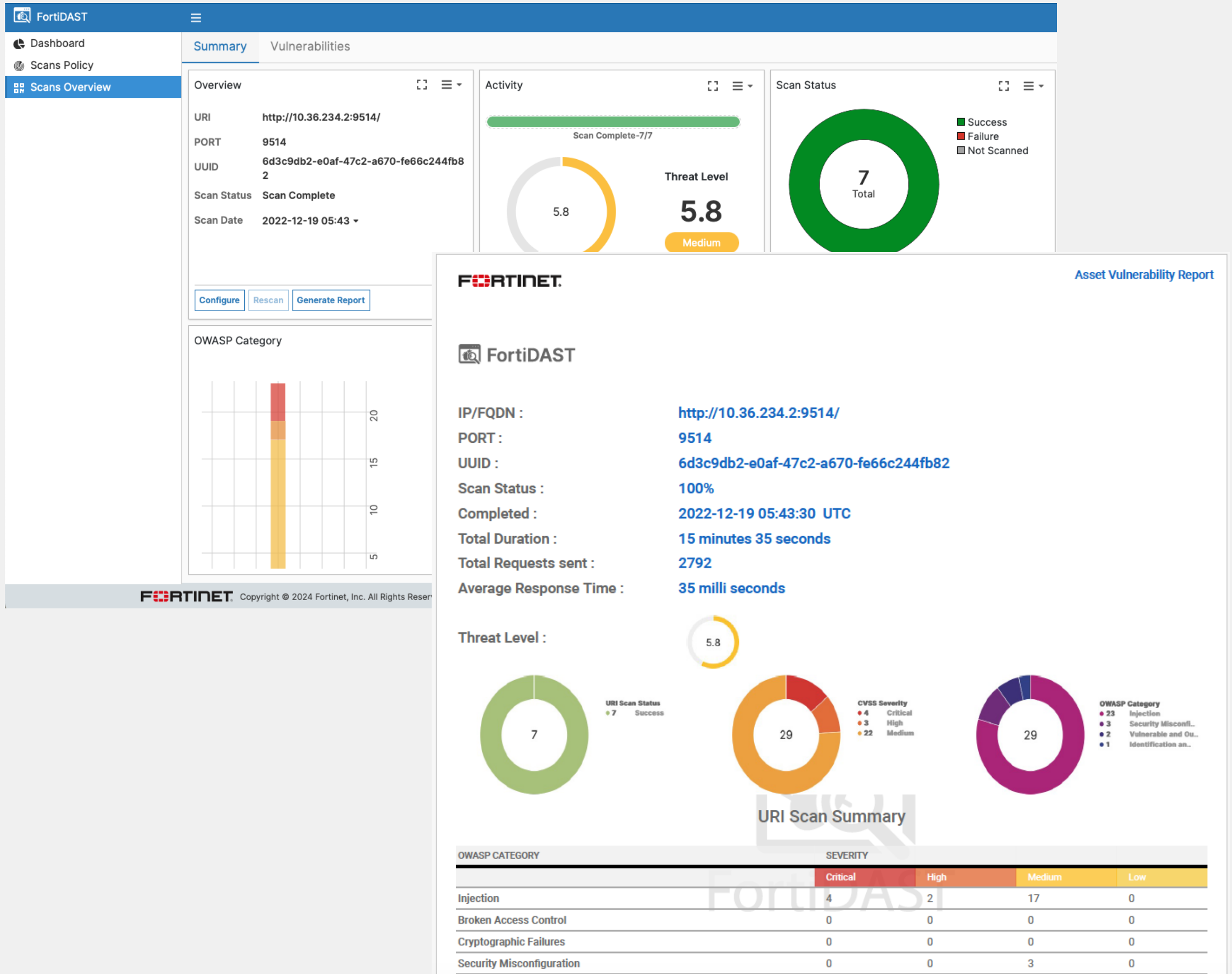
# FortiDevSec DAST Scan Coverage

Comprehensive coverage using FortiDAST (5 app licenses included, stackable)

## Broad Scan Coverage

- Injection (code, LDAP, XSS, SQL etc)
- Broken Access Control (Path Traversal)
- Cryptographic Failures (SSL, weak ciphers etc)
- Security Misconfiguration
- Software & Data Integrity Failures
- Identification and Authentication Bypass
- Vulnerable/Outdated Components

## Comprehensive Results (GUI & Report)



URI http://10.36.234.2:9514/
PORT 9514
UUID 6d3c9db2-e0af-47c2-a670-fe66c244fb82
Scan Status Scan Complete
Scan Date 2022-12-19 05:43

FORTINET

Asset Vulnerability Report

FortiDAST

IP/FQDN : http://10.36.234.2:9514/
PORT : 9514
UUID : 6d3c9db2-e0af-47c2-a670-fe66c244fb82
Scan Status : 100%
Completed : 2022-12-19 05:43:30 UTC
Total Duration : 15 minutes 35 seconds
Total Requests sent : 2792
Average Response Time : 35 milli seconds

Threat Level :

URI Scan Summary

| OWASP CATEGORY | SEVERITY | | | |
|---|---|---|---|---|
| | Critical | High | Medium | Low |
| Injection | 4 | 2 | 17 | 0 |
| Broken Access Control | 0 | 0 | 0 | 0 |
| Cryptographic Failures | 0 | 0 | 0 | 0 |
| Security Misconfiguration | 0 | 0 | 3 | 0 |

# FortiDevSec DAST Scanner - Example

Dynamic Application Security Test - uses FortiDAST (license included)

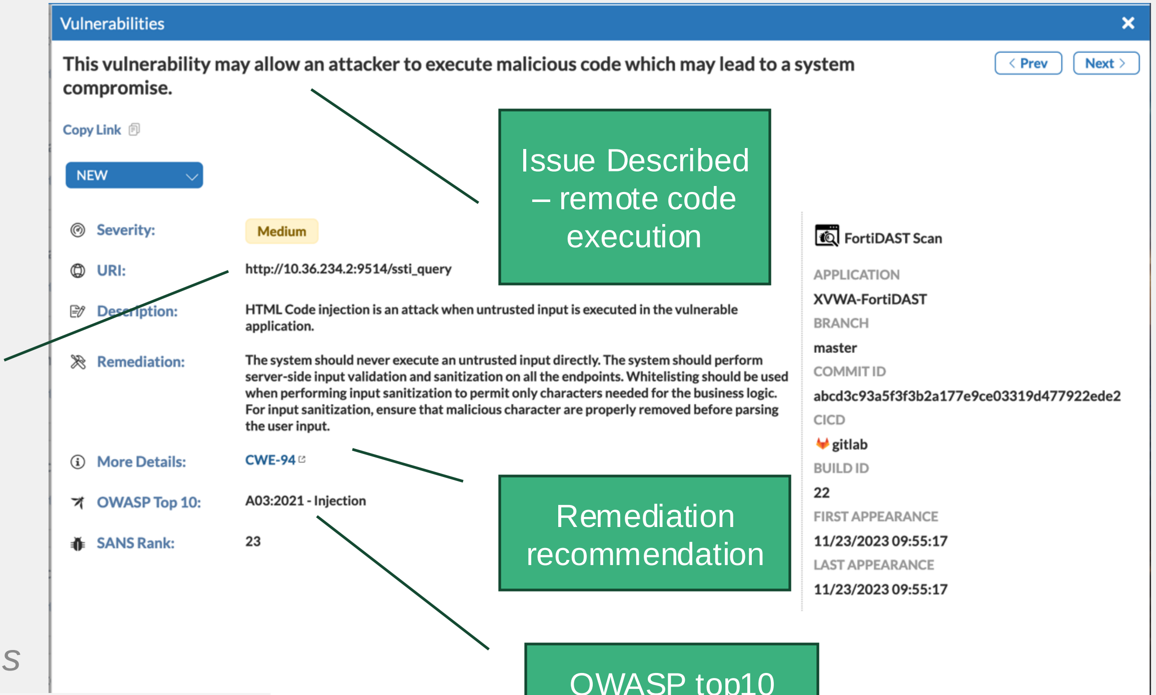
## *Purpose*

Scans a deployed application hosted local/cloud at *runtime* to detect vulnerabilities. Usually done on staging but can be performed in production. Quick and Full scan Available.

### *Sample Result*

Identified real time vulnerabilities for hosted application on IP http://10.36.234.2/URI

*Figure – FortiDevSec DAST Scan result*



Vulnerabilities

This vulnerability may allow an attacker to execute malicious code which may lead to a system compromise.

Severity: Medium

URI: http://10.36.234.2:9514/ssti_query

Description: HTML Code injection is an attack when untrusted input is executed in the vulnerable application.

Remediation: The system should never execute an untrusted input directly. The system should perform server-side input validation and sanitization on all the endpoints. Whitelisting should be used when performing input sanitization to permit only characters needed for the business logic. For input sanitization, ensure that malicious character are properly removed before parsing the user input.

More Details: CWE-94

OWASP Top 10: A03:2021 - Injection

SANS Rank: 23

FortiDAST Scan

APPLICATION
XVWA-FortiDAST

BRANCH
master

COMMIT ID
abcd3c93a5f3f3b2a177e9ce03319d477922ede2

CICD
gitlab

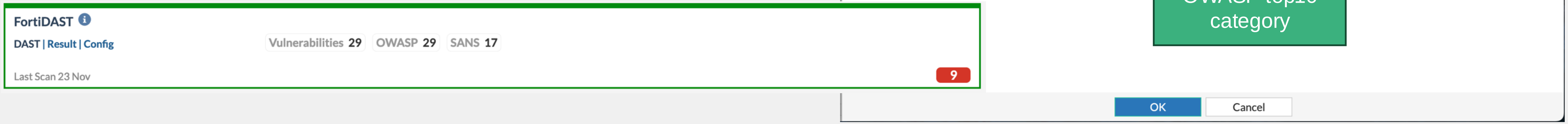
BUILD ID
22

FIRST APPEARANCE
11/23/2023 09:55:17

LAST APPEARANCE
11/23/2023 09:55:17

✕
< Prev
Next >
Copy Link
NEW
Issue Described – remote code execution
URI of application
Remediation recommendation
OWASP top10 category

*FortiDevSec shows DAST vulnerabilities with OWASP and SANs categories*



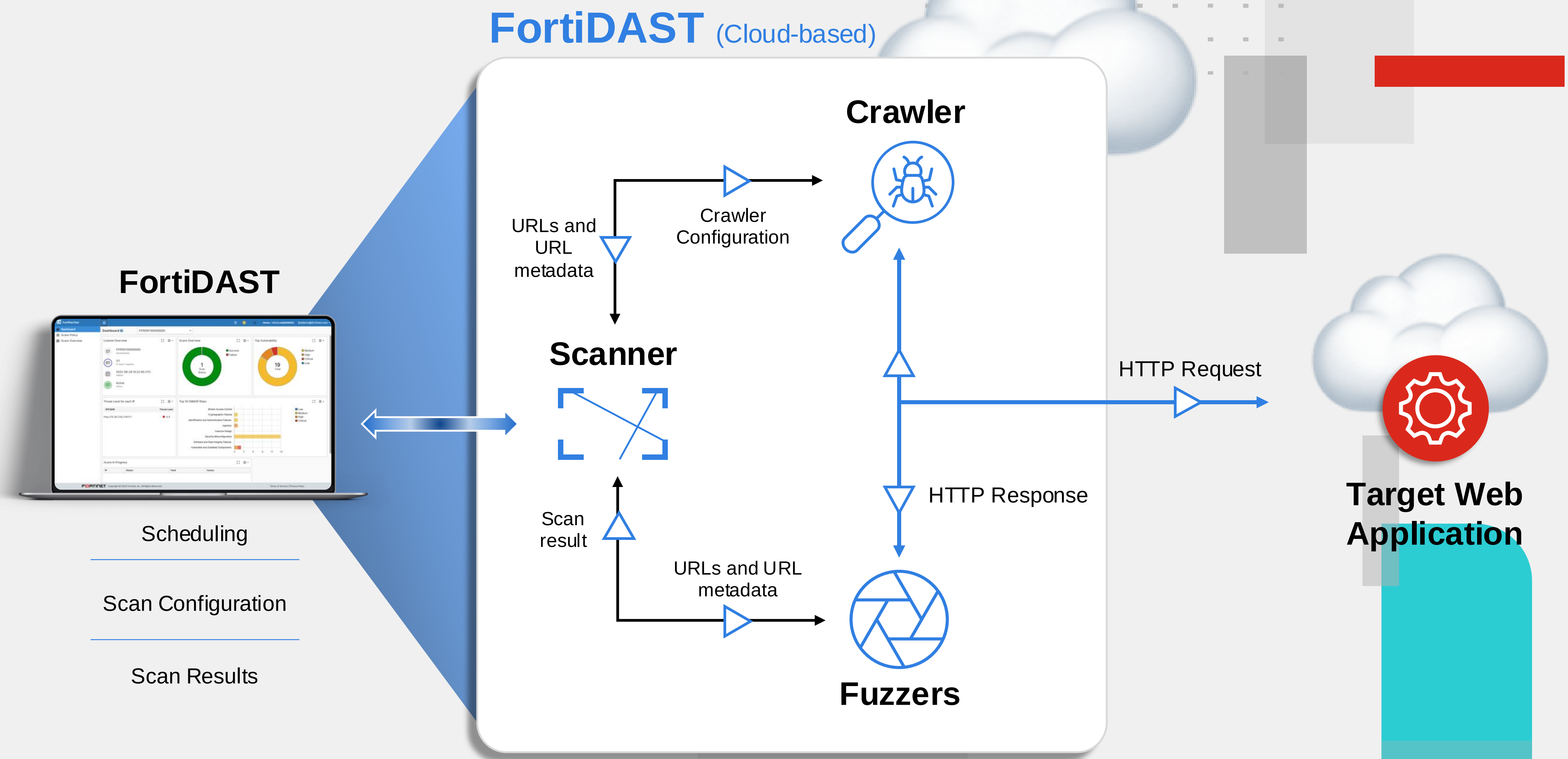
FortiDAST

DAST | Result | Config

Vulnerabilities 29 | OWASP 29 | SANS 17

Last Scan 23 Nov

# How It Works



FortiDAST (Cloud-based)

Crawler

URLs and URL metadata

Crawler Configuration

Scanner

HTTP Request

HTTP Response

Scan result

URLs and URL metadata

Fuzzers

Target Web Application

FortiDAST

Scheduling

Scan Configuration

Scan Results

# FortiDevSec Integrations

# CI/CD Tools Supported by FortiDevSec

Continuous Integration/Continuous Delivery (CI/CD)

Jenkins

GitLab

GitHub

Azure DevOps

AWS CodePipeline

circleci

Google Cloud Build

Bamboo

DRONE

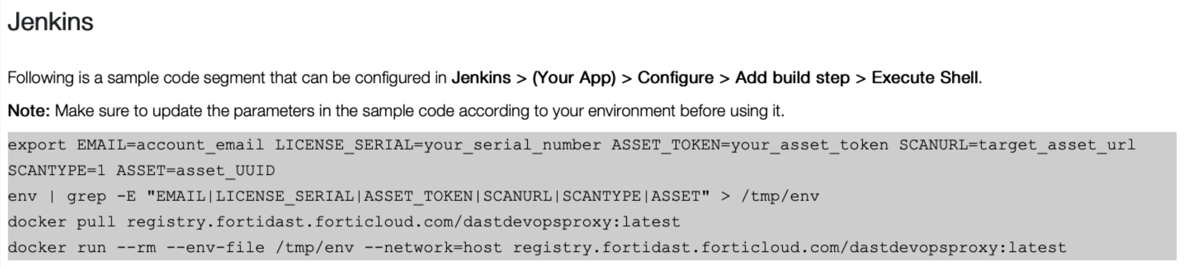
Travis CI

Steps:

1. Developer Copy code segment into CI/CD configurations
2. CI/CD tools download DevSec docker container
3. Container  scans for languages used and download scanners required
4. Only result (and small code snippets) is uploaded to DevSec Cloud

***important** No source code files or libraries will leave customer site!*

*Figure – Jenkins Example*



Jenkins

Following is a sample code segment that can be configured in **Jenkins > (Your App) > Configure > Add build step > Execute Shell**.

**Note:** Make sure to update the parameters in the sample code according to your environment before using it.

```
export EMAIL=account_email LICENSE_SERIAL=your_serial_number ASSET_TOKEN=your_asset_token SCANURL=target_asset_url
SCANTYPE=1 ASSET=asset_UUID
env | grep -E "EMAIL|LICENSE_SERIAL|ASSET_TOKEN|SCANURL|SCANTYPE|ASSET" > /tmp/env
docker pull registry.fortidast.forticloud.com/dastdevopsproxy:latest
docker run --rm --env-file /tmp/env --network=host registry.fortidast.forticloud.com/dastdevopsproxy:latest
```

# FortiDevSec Jira Integration

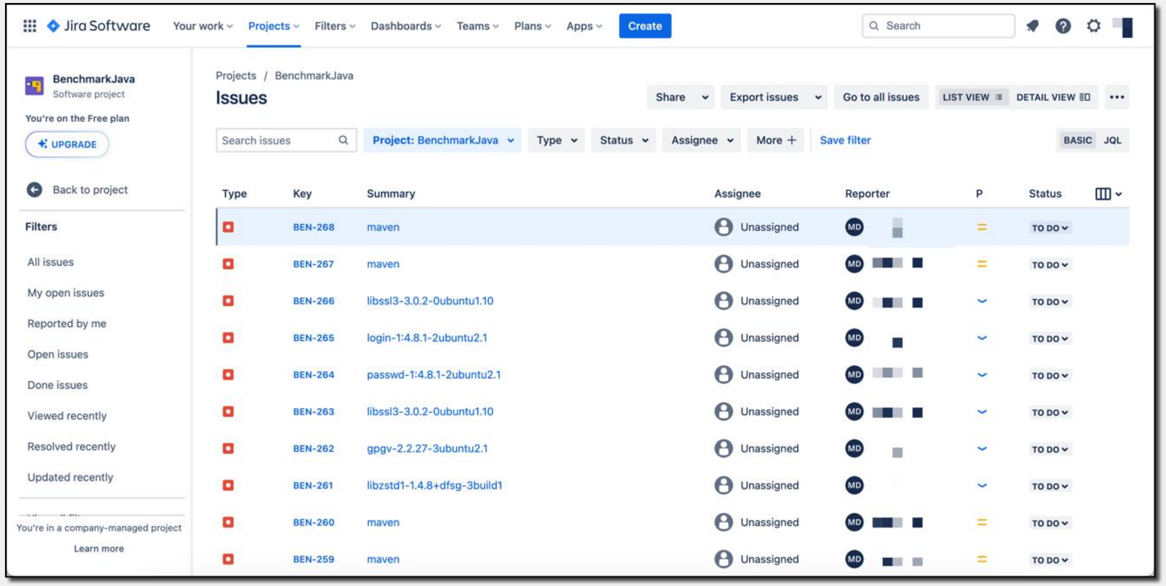
Auto synchronize findings to your own Bug Tracker

## *Purpose*

Allows your teams to re-use their existing workflow to mitigate security issues found by FortiDevSec

## *Support*

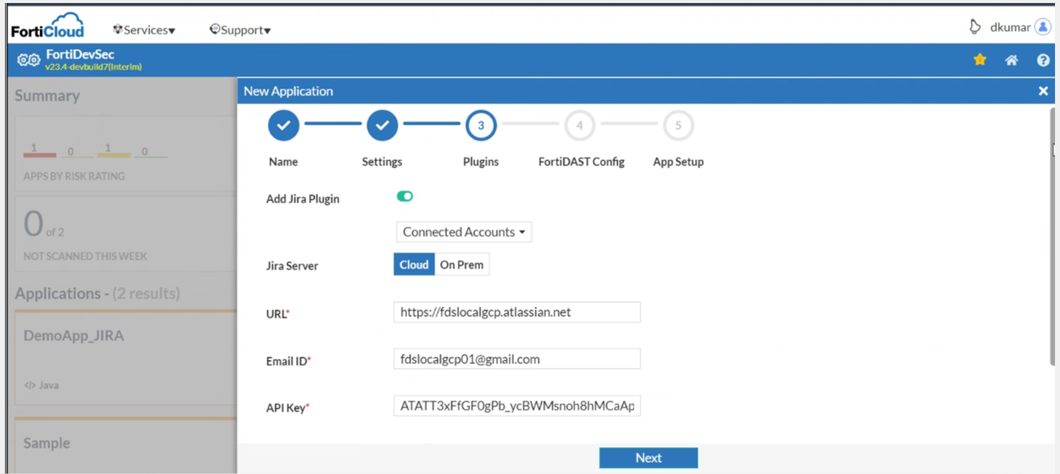
Both on-prem as cloud-based version of Jira is supported



Issues

| Type | Key | Summary | Assignee | Reporter | P | Status |
|---|---|---|---|---|---|---|
| ■ | BEN-268 | maven | Unassigned | MD [redacted] | = | TO DO |
| ■ | BEN-267 | maven | Unassigned | MD [redacted] | = | TO DO |
| ■ | BEN-266 | libssl3-3.0.2-0ubuntu1.10 | Unassigned | MD [redacted] | ﹀ | TO DO |
| ■ | BEN-265 | login-1:4.8.1-2ubuntu2.1 | Unassigned | MD [redacted] | ﹀ | TO DO |
| ■ | BEN-264 | passwd-1:4.8.1-2ubuntu2.1 | Unassigned | MD [redacted] | ﹀ | TO DO |
| ■ | BEN-263 | libssl3-3.0.2-0ubuntu1.10 | Unassigned | MD [redacted] | ﹀ | TO DO |
| ■ | BEN-262 | gpgv-2.2.27-3ubuntu2.1 | Unassigned | MD [redacted] | ﹀ | TO DO |
| ■ | BEN-261 | libzstd1-1.4.8+dfsg-3build1 | Unassigned | MD | ﹀ | TO DO |
| ■ | BEN-260 | maven | Unassigned | MD [redacted] | = | TO DO |
| ■ | BEN-259 | maven | Unassigned | MD [redacted] | = | TO DO |

*Figure – auto create issues in Jira to follow up*



Add Jira Plugin: ●

Jira Server: [✓ Cloud] On Prem

URL*: https://fdslocalgcp.atlassian.net

Email ID*: fdslocalgcp01@gmail.com

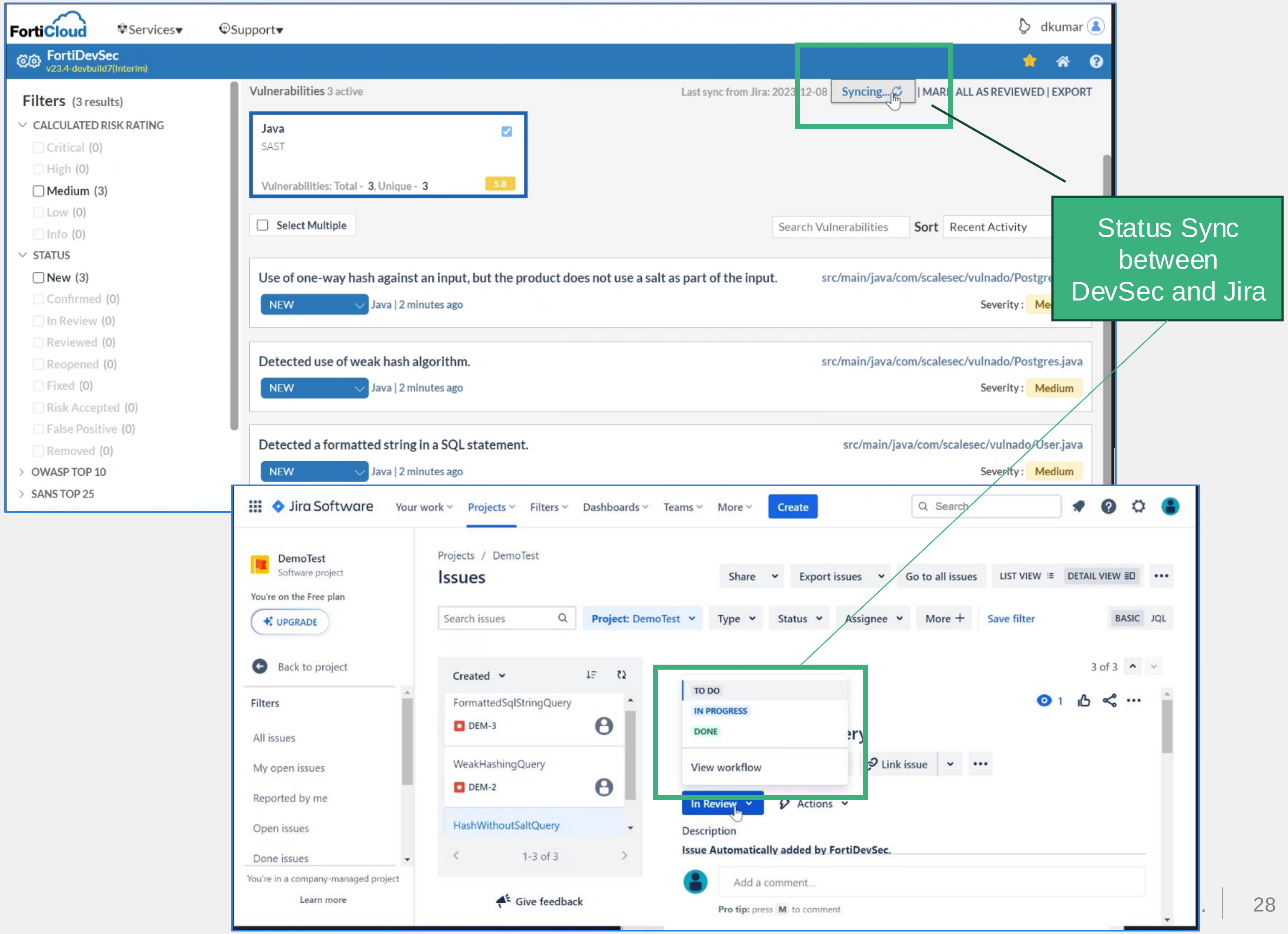
API Key*: ATATT3xFfGF0gPb_ycBWMsnoh8hMCaAp

*Figure – Setup Wizard including Jira Onprem/Cloud configuration*

# Jira Integration – Two Way Synchronization

*Two Way -Synchronization*

- New issues found are added automatically
- Issues found fixed during scan will be removed automatically
- Issues fixed by dev team are synced back to FortiDevSec



Filters (3 results)

CALCULATED RISK RATING
- ☐ Critical (0)
- ☐ High (0)
- ☐ Medium (3)
- ☐ Low (0)
- ☐ Info (0)

STATUS
- ☐ New (3)
- ☐ Confirmed (0)
- ☐ In Review (0)
- ☐ Reviewed (0)
- ☐ Reopened (0)
- ☐ Fixed (0)
- ☐ Risk Accepted (0)
- ☐ False Positive (0)
- ☐ Removed (0)

OWASP TOP 10
SANS TOP 25

Vulnerabilities 3 active

Last sync from Jira: 2023-12-08 Syncing... | MARK ALL AS REVIEWED | EXPORT

Java ☑
SAST
Vulnerabilities: Total - 3, Unique - 3 5.8

☐ Select Multiple — Search Vulnerabilities — Sort Recent Activity

Use of one-way hash against an input, but the product does not use a salt as part of the input. src/main/java/com/scalesec/vulnado/Postgre
NEW | Java | 2 minutes ago — Severity : Me

Detected use of weak hash algorithm. src/main/java/com/scalesec/vulnado/Postgres.java
NEW | Java | 2 minutes ago — Severity : Medium

Detected a formatted string in a SQL statement. src/main/java/com/scalesec/vulnado/User.java
NEW | Java | 2 minutes ago — Severity : Medium

Status Sync between DevSec and Jira

DemoTest
Software project
You're on the Free plan
UPGRADE
Back to project
Filters
All issues
My open issues
Reported by me
Open issues
Done issues
You're in a company-managed project
Learn more

Projects / DemoTest

Issues

FormattedSqlStringQuery
DEM-3

WeakHashingQuery
DEM-2

HashWithoutSaltQuery

1-3 of 3

Give feedback

TO DO
IN PROGRESS
DONE
View workflow

In Review
Actions
Link issue

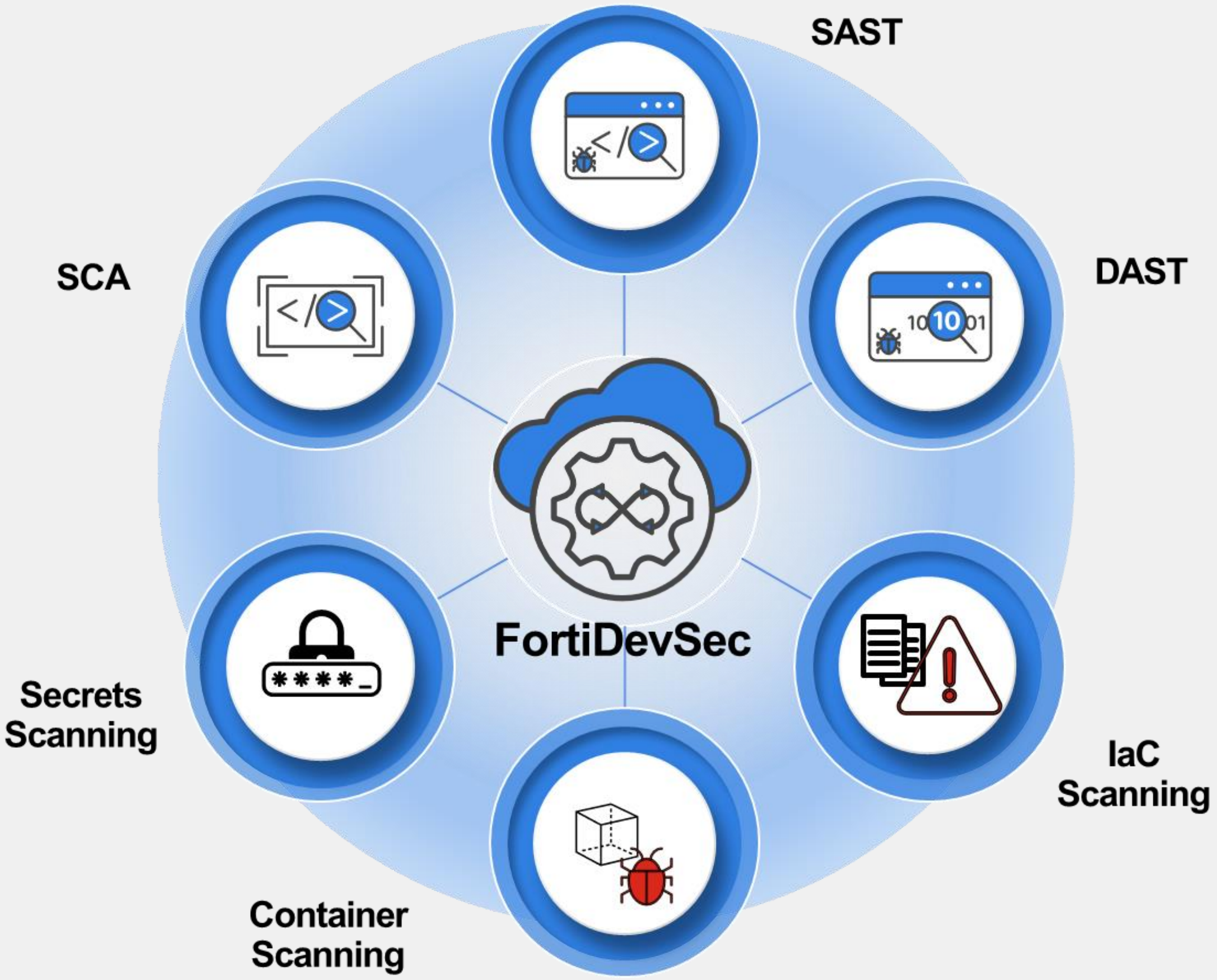
Description

Issue Automatically added by FortiDevSec.

Add a comment...

Pro tip: press M to comment

# One Solution for Comprehensive Application Security Testing



Simple. Focused. Driven.

# FortiWeb

# Challenges of Web/API Security

Cyber threats take advantage of the disruption

## Sophisticated Threats

Endless stream of zero day attacks and application logic attacks that do not have signature protection

## Shadow and Unknown API

Organizations have limited knowledge of their public APIs even though API traffic dominates

## Alert Fatigue

Too many informational, contextless and false positive alerts

# Critical Use Cases

Web Application Protection

## Web Application Security

Protect from OWASP top 10 and other known threats as well as unknown threats.

**OWASP Top 10 Protection With Low FP**

## Protect Internet Facing APIs

Protect the APIs that enable B2B communication and support your mobile applications.

**Discover and Protect APIs**

## Bot Defense

Block the full range of malicious bot activity (content scraping, denial of service, data harvesting, transaction fraud).

**Seamlessly Identify and Block Automated Attacks**

## End Alert Fatigue

Speed up alert investigation and enable SOC analysts to quickly focus on the threats that matter.

**Provide a SOC Analyst Workflow**

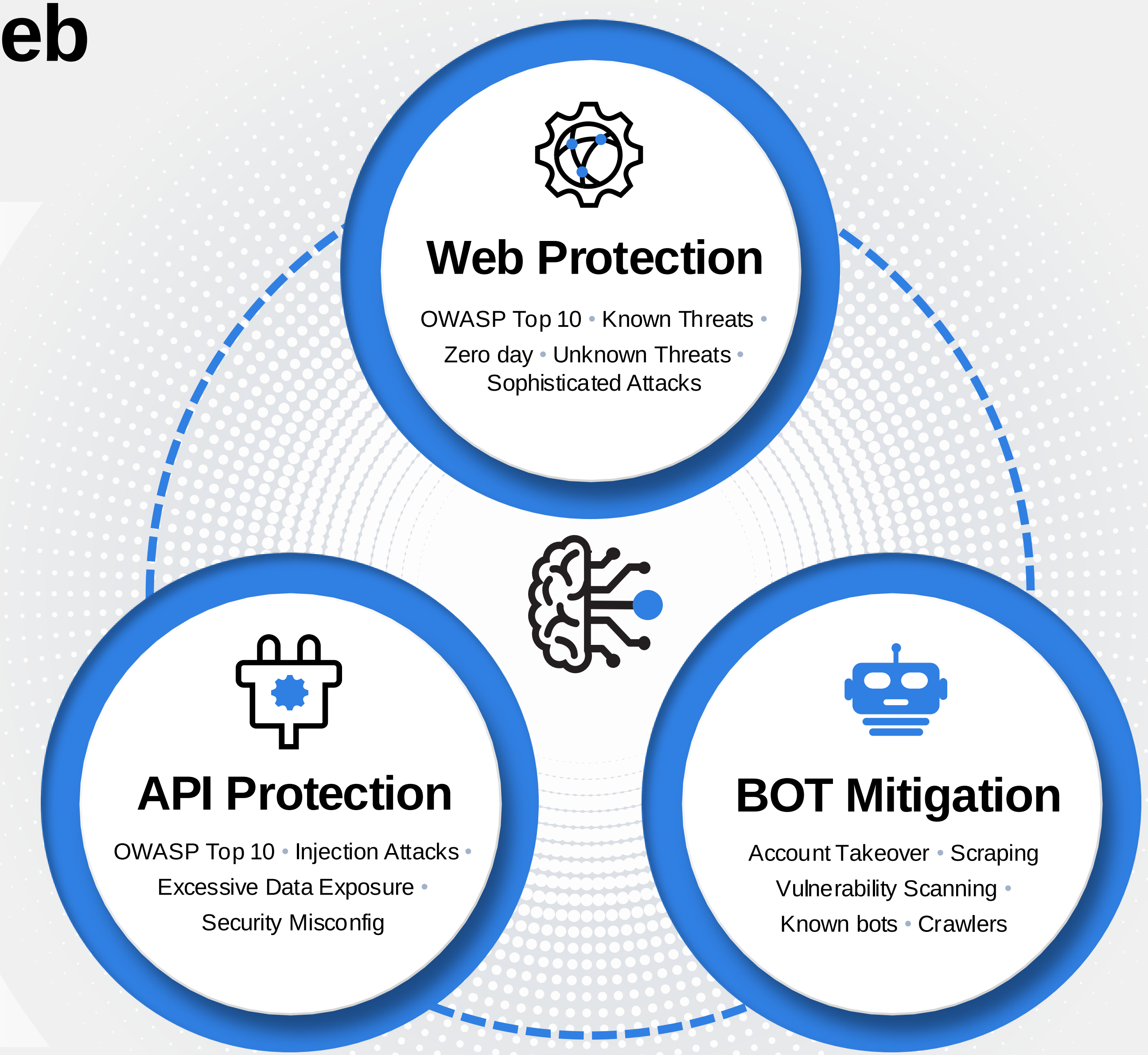
# Introducing FortiWeb

## **Machine Learning Powered**
Web Application & API Security

**Maximum Deployment Flexibility**
SaaS-based, Appliance or VM

**Minimize False Positives**
Sophisticated techniques to reduce false positives

**Threat Analytics** addresses alert fatigue and speeds up alert security investigation



### Web Protection

OWASP Top 10 • Known Threats • Zero day • Unknown Threats • Sophisticated Attacks

### API Protection

OWASP Top 10 • Injection Attacks • Excessive Data Exposure • Security Misconfig

### BOT Mitigation

Account Takeover • Scraping • Vulnerability Scanning • Known bots • Crawlers

# FortiWeb

MACHINE LEARNING

## API Discovery and Protection

API Discovery using URL clustering with schema awareness, automatic schema generation, schema enforcement

## Threat Analytics

Analyze million of events using ML to identify common characteristics and patterns and group them into meaningful security incidents

## Web Protection

Zero day attack protection using two layer solution (HMM and SVM), Anomaly verification, continuous learning

## Bot Mitigation

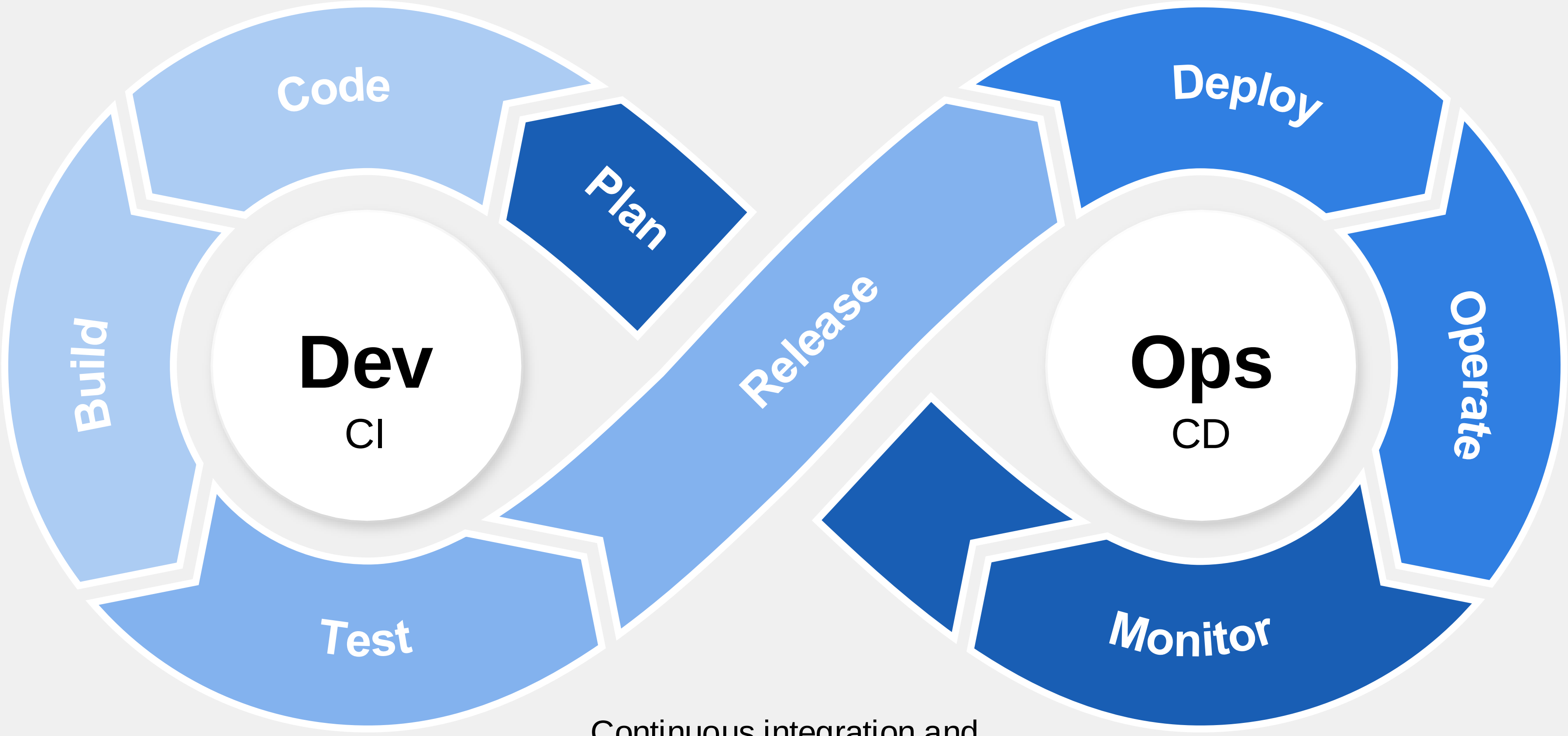
Behavioral learning using ML SVM based on 13 different traffic dimensions, automated verification using training samples

# Why Machine Learning?

# Machine Learning for Web Application Protection

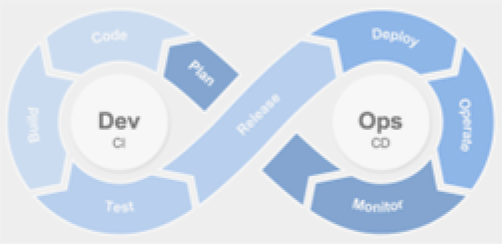
Reduce friction when deploying web applications

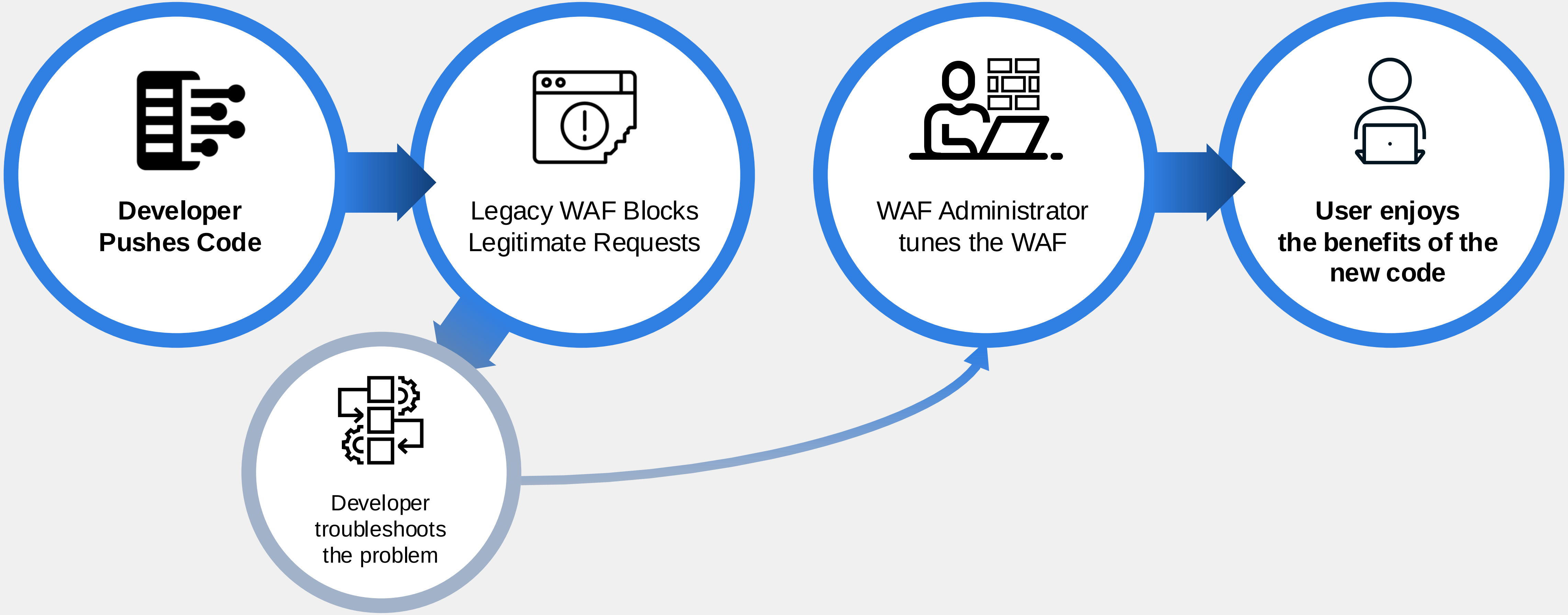
## Why Machine Learning for Web Application Protection Matters for Customers



Continuous integration and
continuous deployment (CI/CD)

# Old Fashioned WAFs add friction





**Developer Pushes Code**

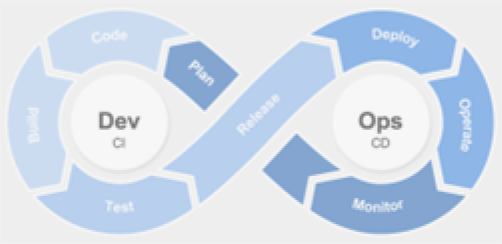
Legacy WAF Blocks Legitimate Requests

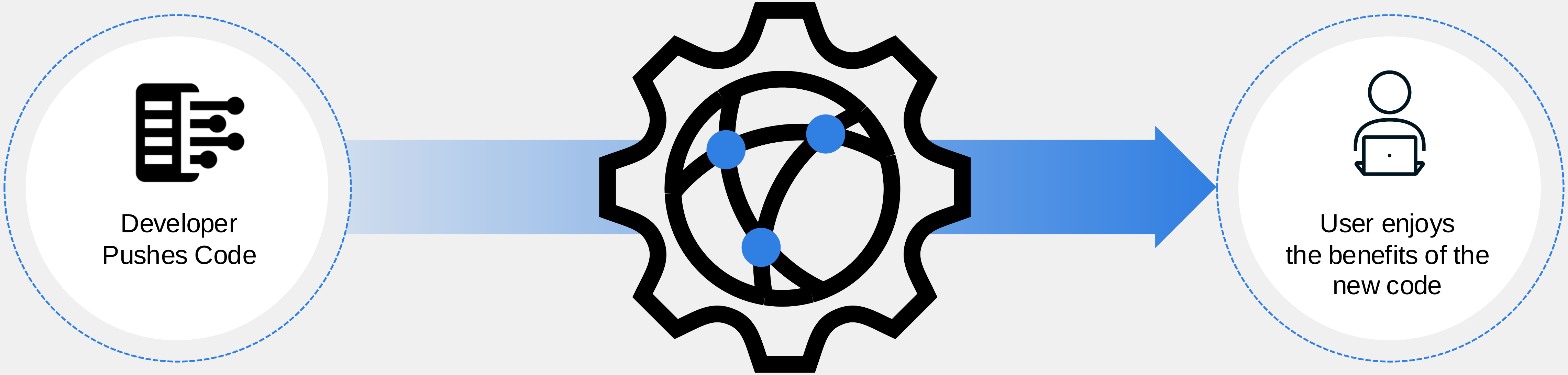
Developer troubleshoots the problem

WAF Administrator tunes the WAF

**User enjoys the benefits of the new code**

# Machine Learning for Web Application Protection

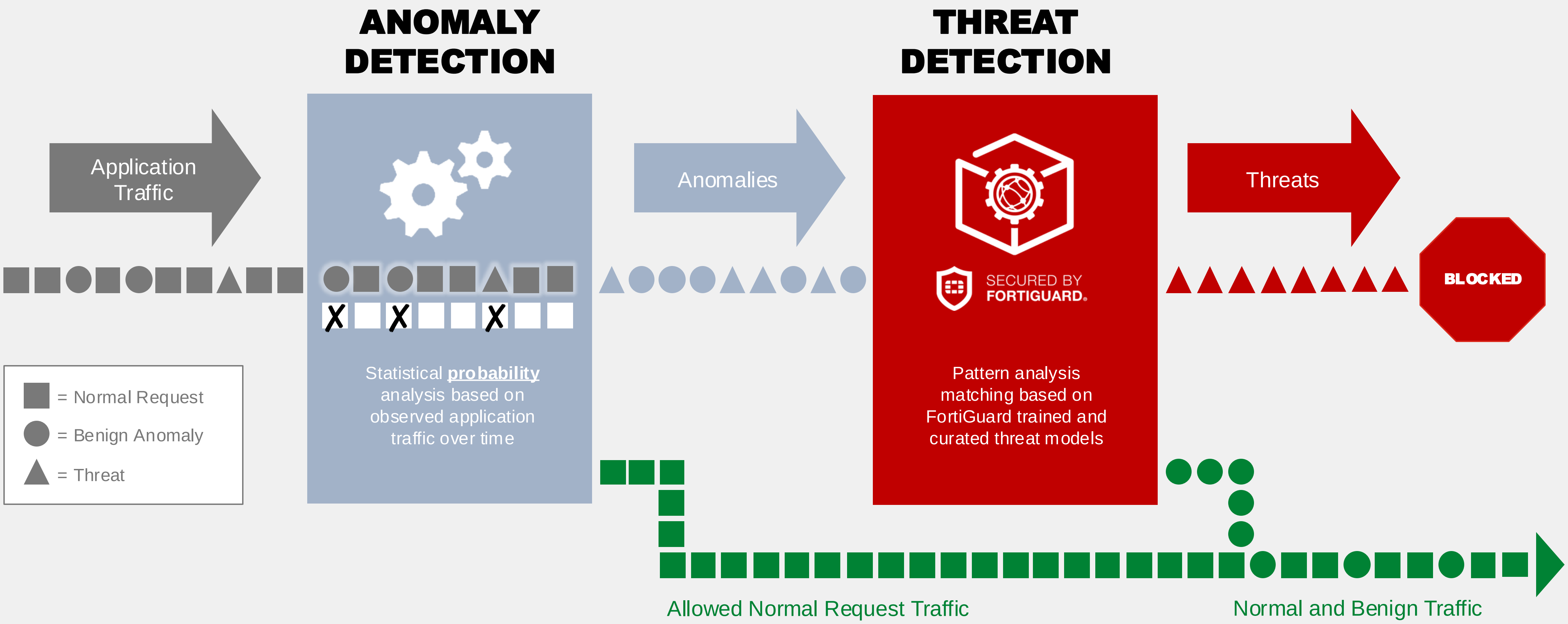




Developer
Pushes Code

User enjoys
the benefits of the
new code

# FortiWeb Employs 2 Layers of Machine Learning



■ = Normal Request

● = Benign Anomaly

▲ = Threat

Statistical **probability** analysis based on observed application traffic over time

Pattern analysis matching based on FortiGuard trained and curated threat models

**Reduce friction when deploying web applications!**

# Web Protection - Anomaly Detection Layer I

## GOAL:

- Build a profile of allowed behavior that represents the application's true state
- Trigger anomalies when requests violate probability
- Automatically and immediately update profile when application state changes

## FortiWeb ML

- Builds mathematical models with just 400 samples (Uses Hidden Markov Model (HMM) algorithm
- Continuously builds new mathematical models as more samples are collected
- Addresses incomplete profiles
- Addresses application changes

### Collect

- Gather samples
- Minimum 400
- Continuously collect additional samples

### Build

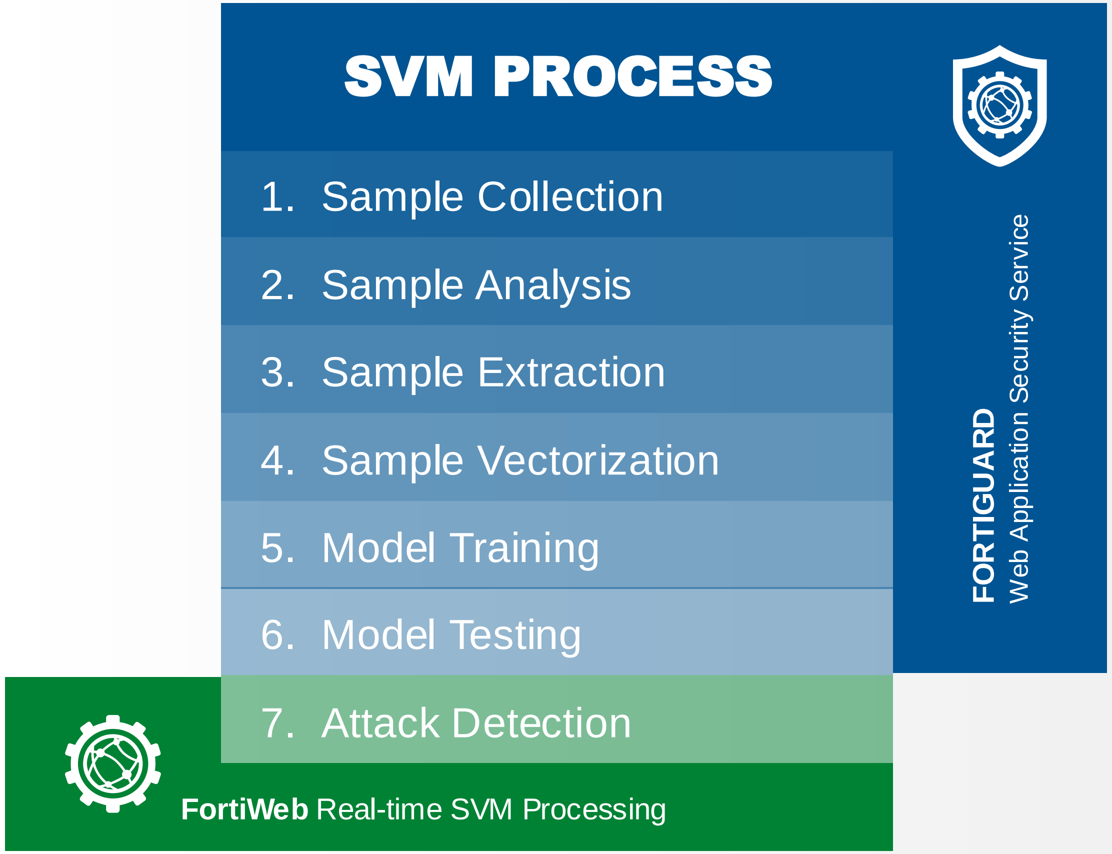
- Build mathematical models
- Establish parameters
- Set variances
- Continuously evaluate enhanced models

### Run

- Enforce Action, trigger anomaly

# Web Protection - Anomaly Detection Layer II

- 2nd layer Machine Learning using Support Vector Machine (SVM) algorithm
- FortiWeb uses threat models trained using thousands of attack samples to identify new attacks
- Every anomaly is tested against the threat models
- Unlike traditional signatures (regex), SVM learns attack model elements so can cover variations of attack
- FortiGuard continuously pushes to FortiWeb updated threat models
- “Heavy lifting” done by FortiGuard
- Minimal performance impact to FortiWeb



**SVM PROCESS**

1. Sample Collection
2. Sample Analysis
3. Sample Extraction
4. Sample Vectorization
5. Model Training
6. Model Testing
7. Attack Detection

**FORTIGUARD** Web Application Security Service

**FortiWeb** Real-time SVM Processing

# API Discovery and Protection

**APIs are developed and managed differently than standard web applications**

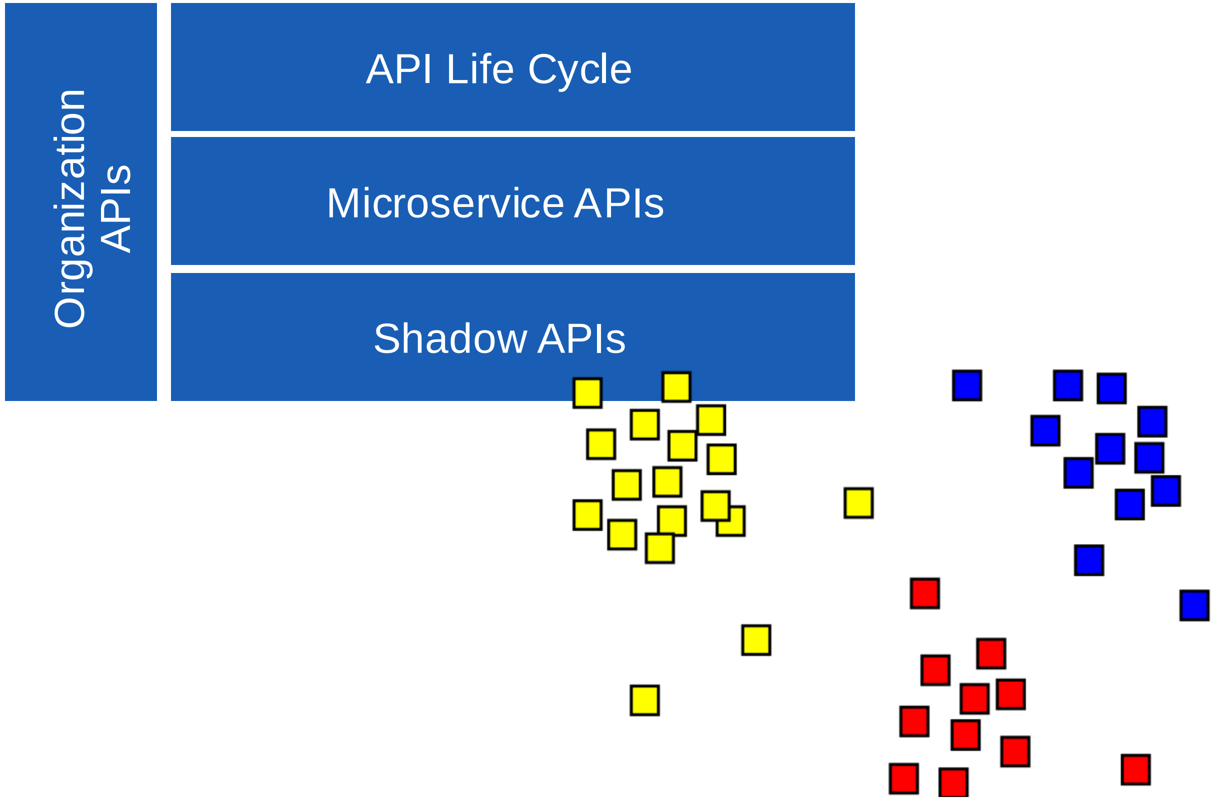
- Shadow APIs – developed as part of the app implementation, not known publicly
- Microservices introduce many internal APIs
- API lifecycle – API evolution/deprecation/temporality
- You can't secure what you don't know.
  API visibility is key

**API Discovery**

- Identify all API endpoints
- Identify which APIs include PII

**API Protection**

- Restores the API specification from user behavior
- FortiWeb uses machine learning based URL clustering with schema awareness algorithms
- Clustering is grouping data points based on similarities and patterns

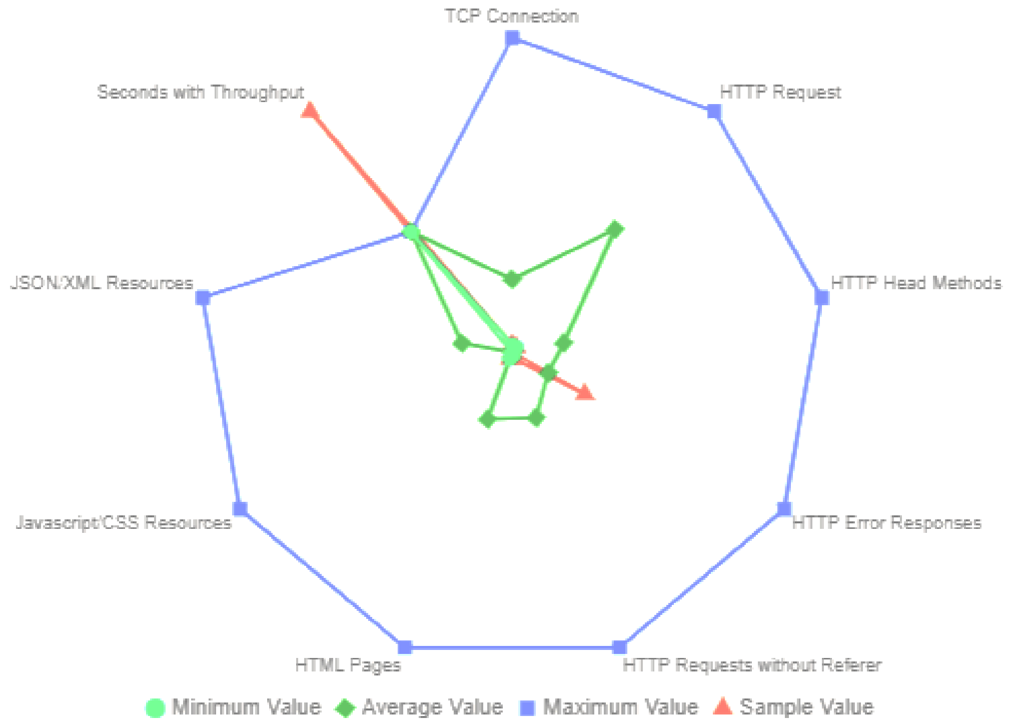


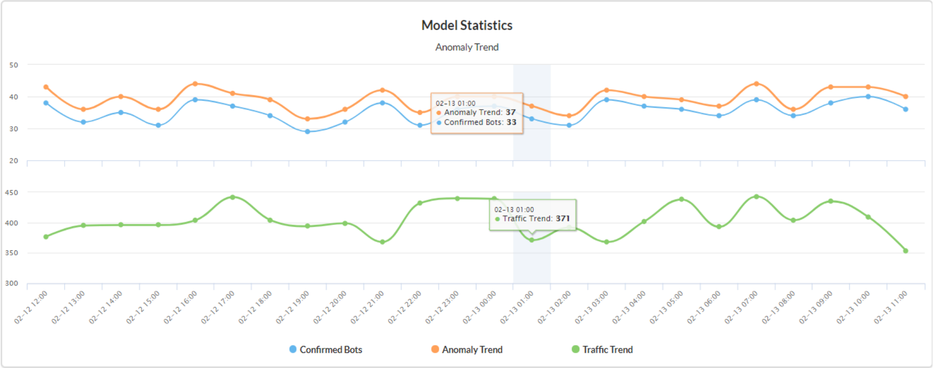
# Bot Mitigation

- 30%-50% of internet traffic is automated traffic. Some industries hit harder (travel, e-commerce, real estate)
- Bad bots involved in scraping, fraud, competitive data mining, personal and financial data harvesting, ticketing, account take over, carding, spam and more
- Bot sophistication varies from dumb, easy to identify (25%) to sophisticated, human like bots (20%)

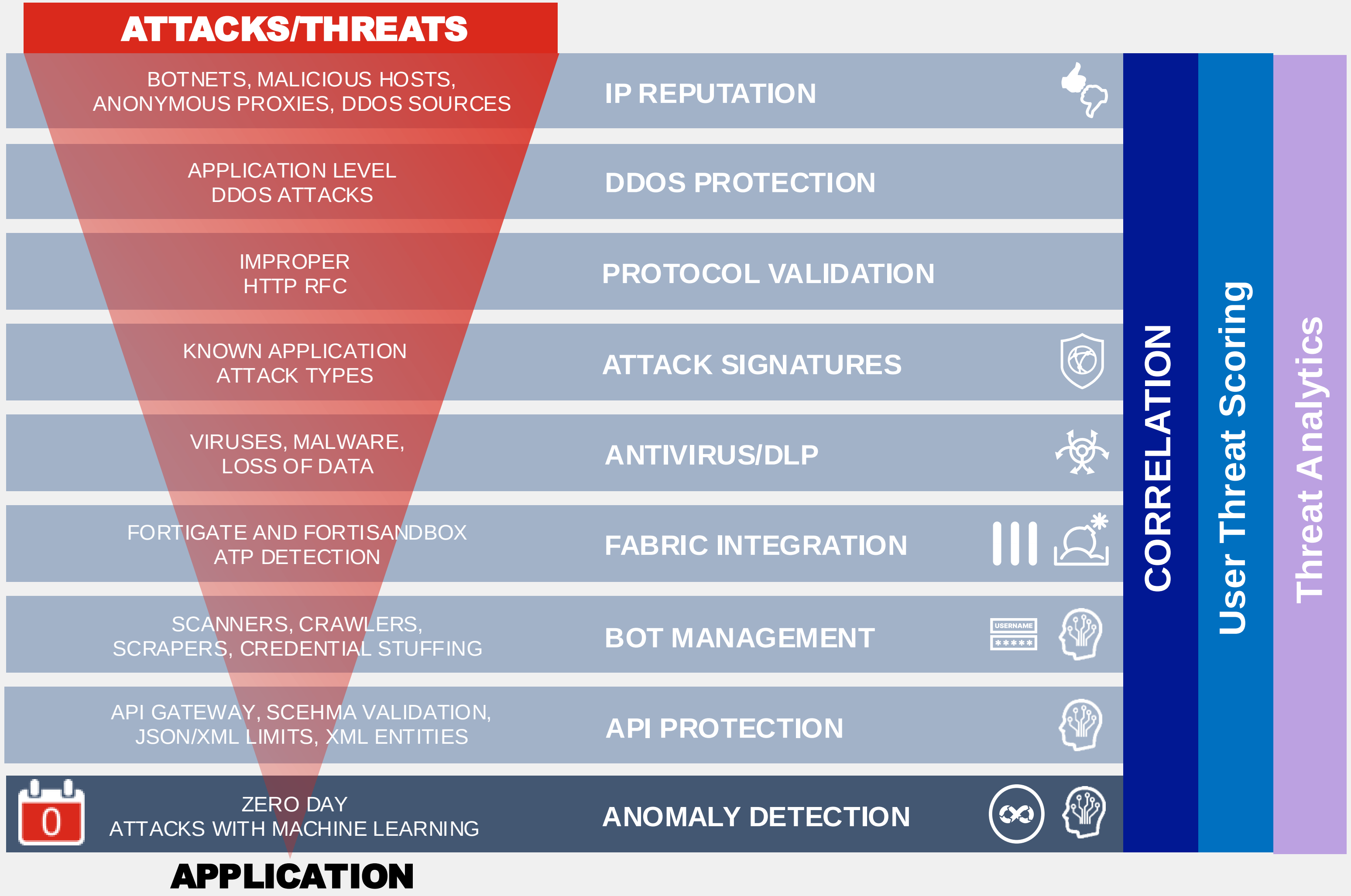
## FortiWeb ML

- Uses one class SVM algorithm
- Validates each sample using JS
- Builds models across 13 traffic dimensions
- Verifies models with additional test samples





# Protection Across all Layers



| ATTACKS/THREATS | |
|---|---|
| BOTNETS, MALICIOUS HOSTS, ANONYMOUS PROXIES, DDOS SOURCES | IP REPUTATION |
| APPLICATION LEVEL DDOS ATTACKS | DDOS PROTECTION |
| IMPROPER HTTP RFC | PROTOCOL VALIDATION |
| KNOWN APPLICATION ATTACK TYPES | ATTACK SIGNATURES |
| VIRUSES, MALWARE, LOSS OF DATA | ANTIVIRUS/DLP |
| FORTIGATE AND FORTISANDBOX ATP DETECTION | FABRIC INTEGRATION |
| SCANNERS, CRAWLERS, SCRAPERS, CREDENTIAL STUFFING | BOT MANAGEMENT |
| API GATEWAY, SCEHMA VALIDATION, JSON/XML LIMITS, XML ENTITIES | API PROTECTION |
| ZERO DAY ATTACKS WITH MACHINE LEARNING | ANOMALY DETECTION |

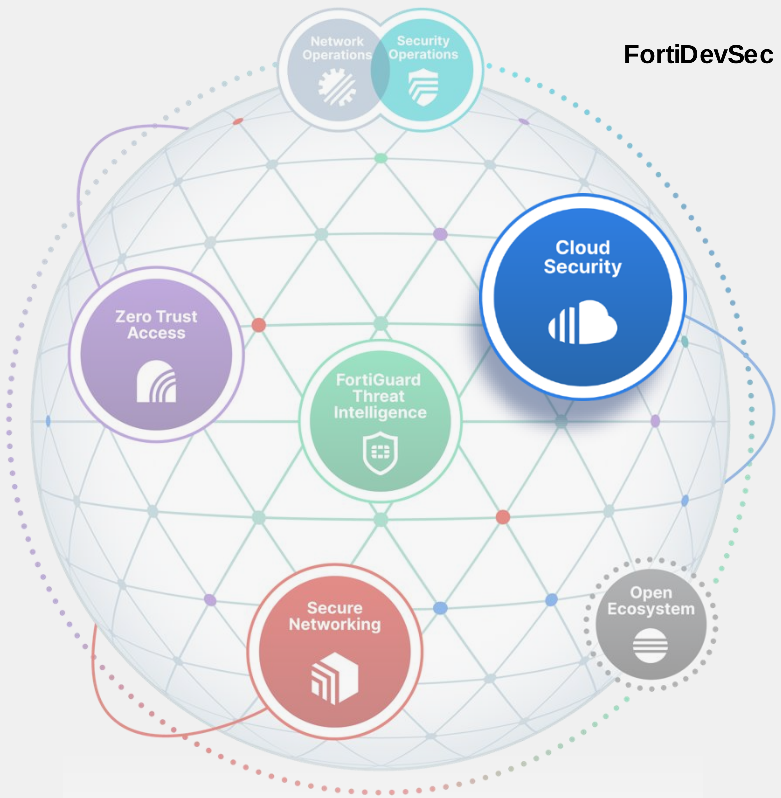
APPLICATION

CORRELATION

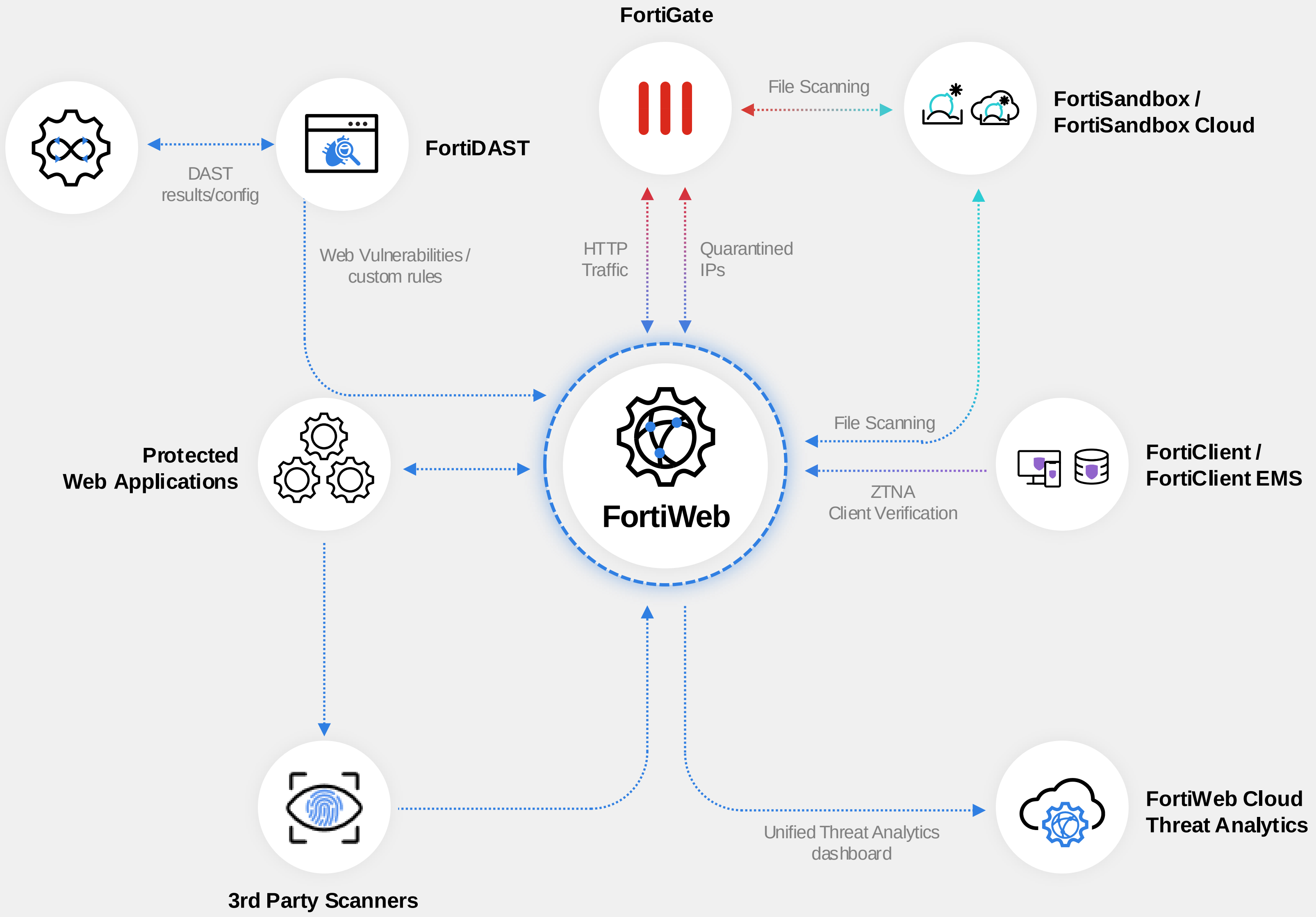
User Threat Scoring

Threat Analytics

# Security Fabric Integrations



A FortiWeb can be configured to join a Security Fabric through the root or downstream FortiGate.

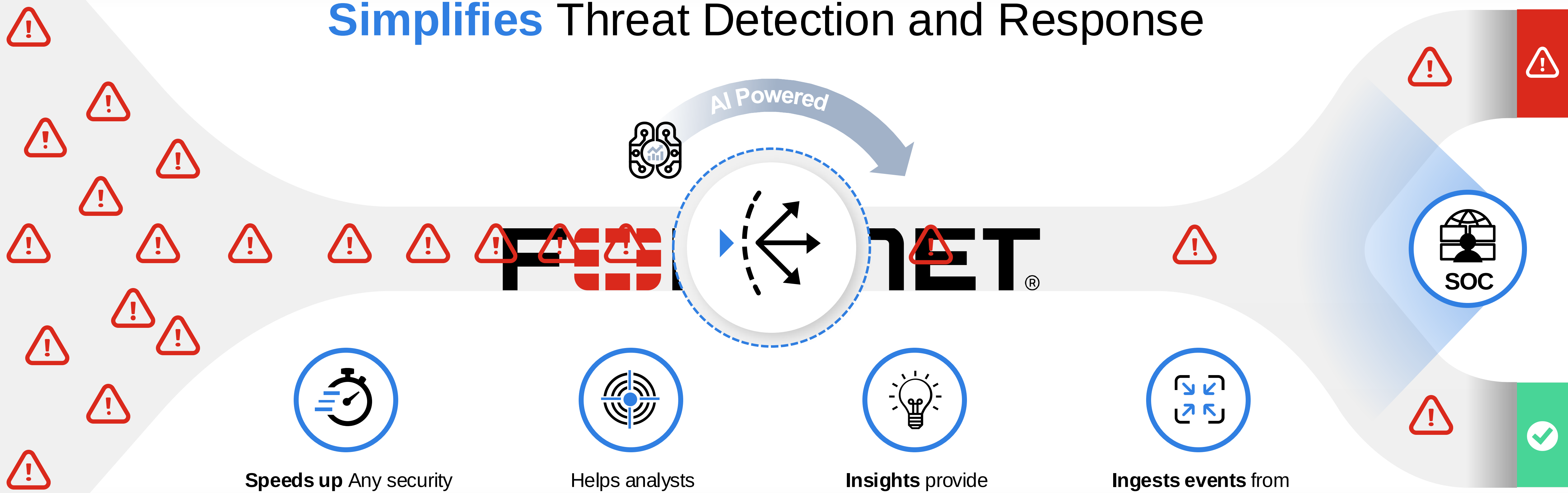


FortiDevSec
DAST results/config
FortiDAST
FortiGate
File Scanning
FortiSandbox / FortiSandbox Cloud
Web Vulnerabilities / custom rules
HTTP Traffic
Quarantined IPs
Protected Web Applications
FortiWeb
File Scanning
ZTNA Client Verification
FortiClient / FortiClient EMS
3rd Party Scanners
Unified Threat Analytics dashboard
FortiWeb Cloud Threat Analytics

# Threat Analytics

# Threat Analytics

## Simplifies Threat Detection and Response



**Speeds up** Any security alerts investigation

Helps analysts **focus** on the most important threats - alleviates alert fatigue

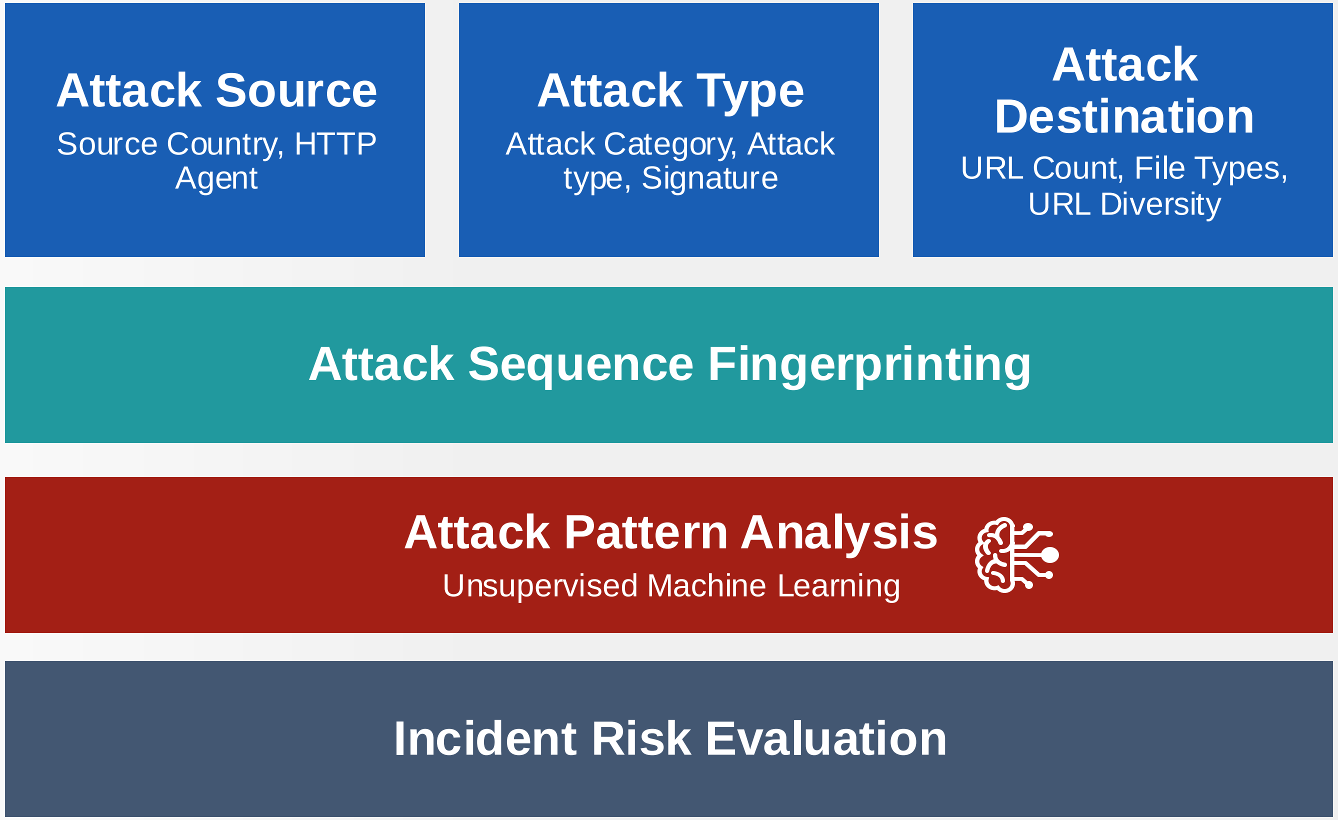
**Insights** provide suggestions to harden security based on findings

**Ingests events** from across your entire hybrid cloud environments

# How it Works

**FortiWeb Threat Analytics** uses machine learning algorithms to identify attack patterns and aggregate them into security incidents across customer entire application assets.

- Aggregate attacks into sequences
  - Same source and destination
  - No match for 60 min
- Create fingerprints for attack sequences
- Use ML to identify patterns in fingerprints
- Aggregate sequences into incidents
- Evaluate incident risk. Severity is impacted by –
  - Severity of every attack in incident
  - Number of attacks in incident
  - Variety of attack types



**Attack Source**
Source Country, HTTP Agent

**Attack Type**
Attack Category, Attack type, Signature

**Attack Destination**
URL Count, File Types, URL Diversity

**Attack Sequence Fingerprinting**

**Attack Pattern Analysis**
Unsupervised Machine Learning

**Incident Risk Evaluation**

# Summing it all up

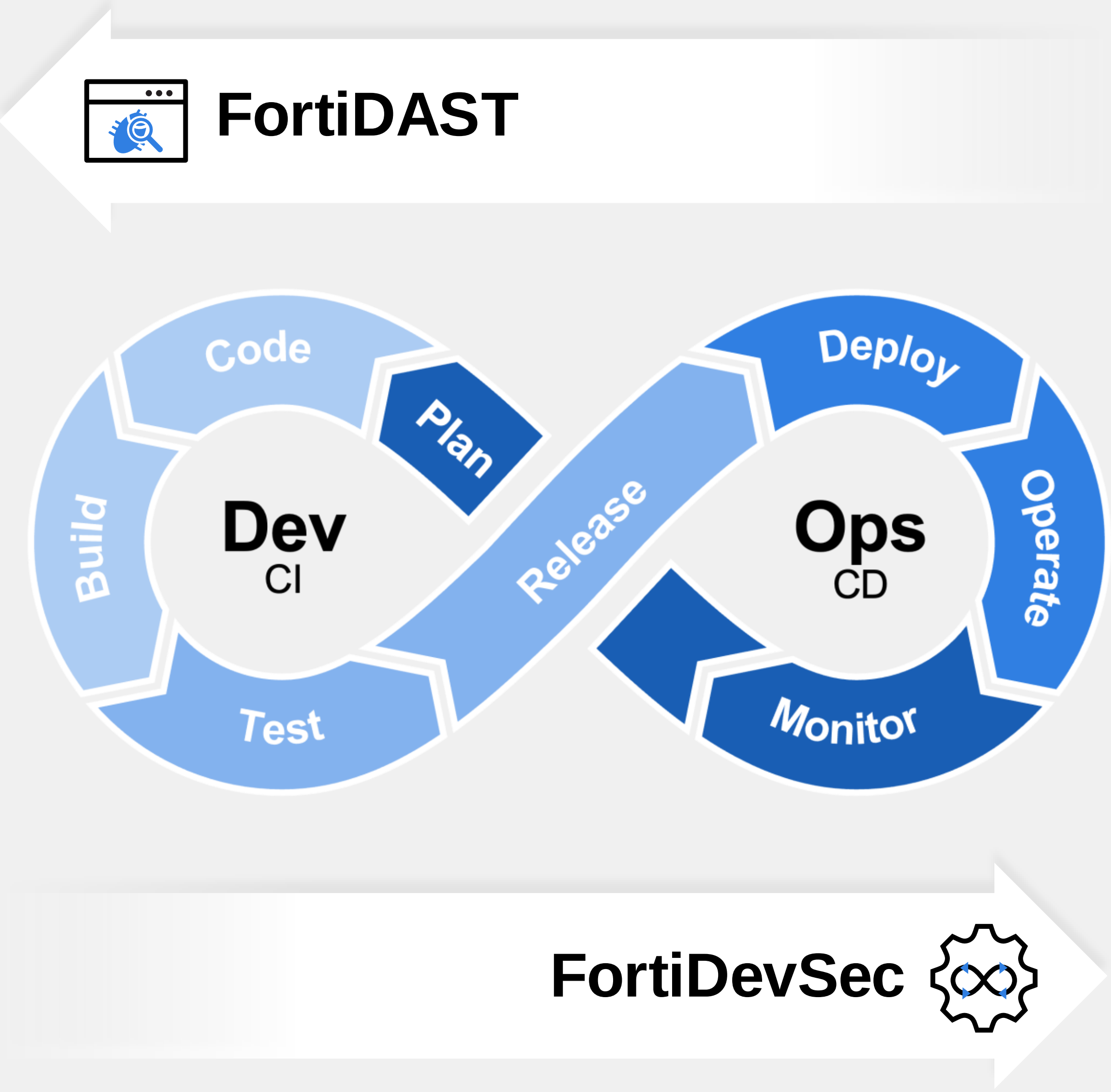
# Web App Security from Dev to Prod

**Black-Box Dynamic Application Security Testing**

- Automated Vulnerability Scanning
- Advanced Crawler
- Fuzzer Expertise
- Detailed and Summary Reporting

**DevOps-first Application Security Testing**

- Simplifies AppSec for Modern DevOps
- Comprehensive Vulnerability Management
- Noise reduction



FortiWeb

**Machine Learning enhanced Web Application and API Protection**

- Web Application Security
- Protect Internet Facing APIs
- Bot Defense
- End Alert Fatigue