



DevSecOps Day EMEA

Unlock 2024

Shielding the Foundation

Security Across Your Software Supply Chain

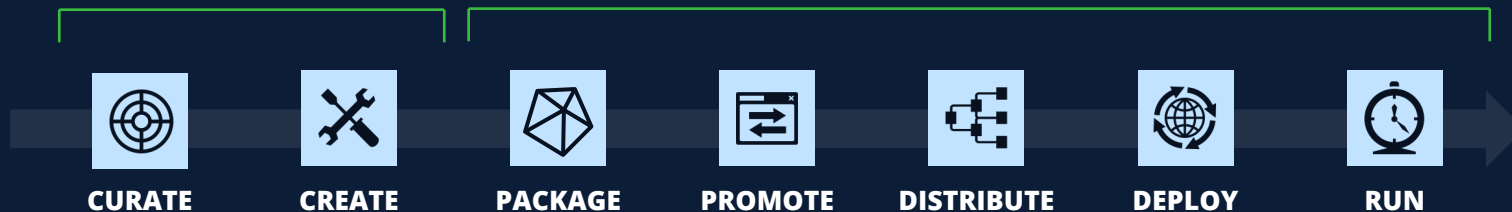
Stephen Chin, VP of Developer Relations

@JFrog



CODE

BINARIES



THE WORLD RUNS ON OSS & 3RD-PARTY COMPONENTS
ACROSS EVERY STAGE OF THE SDLC



CODE

BINARIES



SOFTWARE SUPPLY CHAIN ATTACKS
ARE UP 100x

DEVELOPERS
ARE THE CLEAR TARGETS

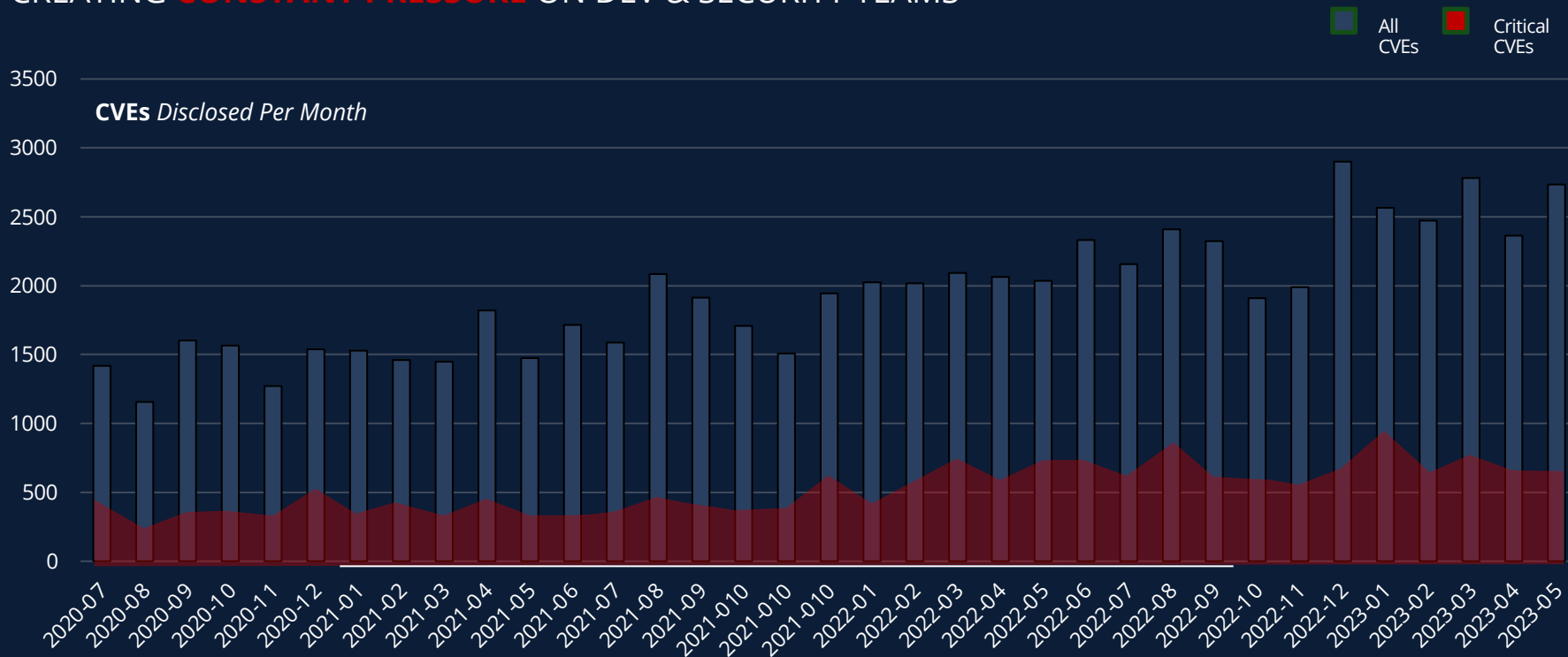


BUT THEY ARE
TIRED OF SECURITY



THE RATE OF **PUBLISHED CVEs** IS INCREASING

CREATING **CONSTANT PRESSURE** ON DEV & SECURITY TEAMS



YET
CRITICAL CVEs
IN COMMON COMPONENTS

CAN HAVE
NO REAL SECURITY IMPACT

Yet **CRITICAL CVEs** in common components can have **NO REAL SECURITY IMPACT**


“ Curl is used
daily by virtually
**EVERY INTERNET-USING
HUMAN** on the planet ”

curl://



OVER **20
BILLION**
INSTALLATIONS

YET **CRITICAL CVEs** IN COMMON COMPONENTS CAN HAVE NO REAL SECURITY IMPACT

 daniel:// stenberg://
@bagder

CVE-2020-19909 is everything

All made up.

Another 9.8 CRITICAL curl problem. All made up.

daniel.haxx.se/blog/2023/08/2...

4 long days later....

Base Score: **9.8 CRITICAL**

4:03 PM · Aug 25, 2023 · 338.3K Views

390 Reposts 41 Quotes 1,271 Likes 123 Bookmarks

Current Description

**** DISPUTED **** Integer overflow vulnerability in tool_operate.c in curl 7.65.2 via a large value as the retry delay. NOTE: many parties report that this has no direct security impact on the curl user browser, it may (in theory) cause a denial of service to associated systems or networks if, for example, --retry-delay is set to a value much smaller than what was intended. This is not especially plausible because the overflow only happens if the user was told to specify that curl should wait weeks (or longer) before trying to recover from a transient error.

MANY CRITICAL CVEs IN COMMON COMPONENTS ARE NON-EXPLOITABLE IN 99% OF CASES



120,000 BINARIES
0 EXPLOITABLE CASES

🚩 CVE-2023-20873 Detail

MODIFIED

This vulnerability has been modified since it was last analyzed by the NVD. It is awaiting reanalysis which may result in further changes to the information provided.

Description

In Spring Boot versions 3.0.0 - 3.0.5, 2.7.0 - 2.7.10, and older unsupported versions, an application that is deployed to Cloud Foundry could be susceptible to a security bypass. Users of affected versions should apply the following mitigation: 3.0.x users should upgrade to 3.0.6+. 2.7.x users should upgrade to 2.7.11+. Users of older, unsupported versions should upgrade to 3.0.6+ or 2.7.11+.

Severity

CVSS Version 3.x

CVSS Version 2.0

CVSS 3.x Severity and Metrics:



NIST: NVD

Base Score:

9.8 CRITICAL

Vector:

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H



AND WHILE DEVELOPERS ARE **SWAMPED**,
ATTACKERS ARE COMING UP WITH
NEW ATTACKS



SCORE <JFROG SECURITY>

1337



ABUSING **SECRETS** IN BINARIES LEAKED TO PUBLIC REPOSITORIES



PRIVATE REPOSITORY

LEAK



tescodesign
1.0.0 • Public • Published 13 days ago

Code

```
<!-- Send Grid -->  
<add key="SendGridApiKey" value="SG.1DK3eer3SySpKM35yh12Xg.1B1"
```

NON-COMPANY ACCOUNT ON PUBLIC REPOSITORY

OVER 250K TOKENS DETECTED BY JFROG!

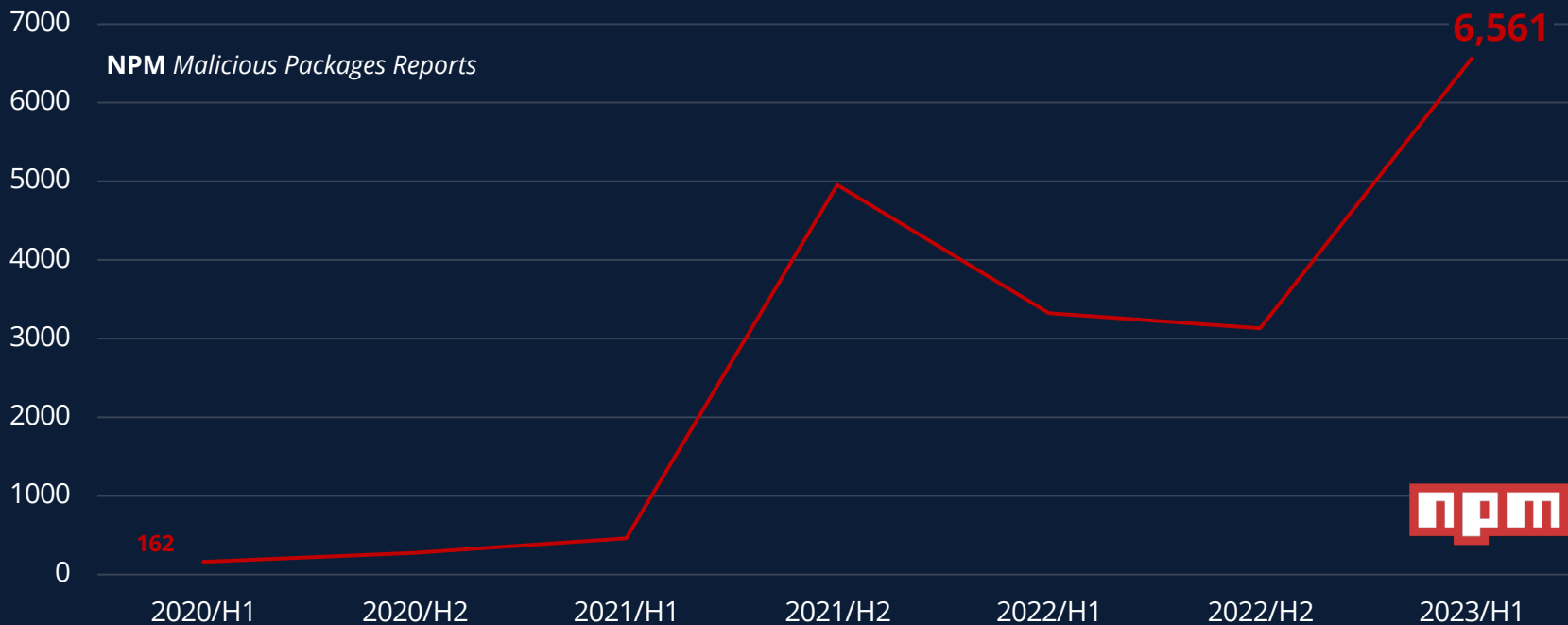
slack

GitHub


Read All Messages


Read & Modify Source Code

THE RATE OF **MALICIOUS PACKAGE ATTACKS** IS INCREASING

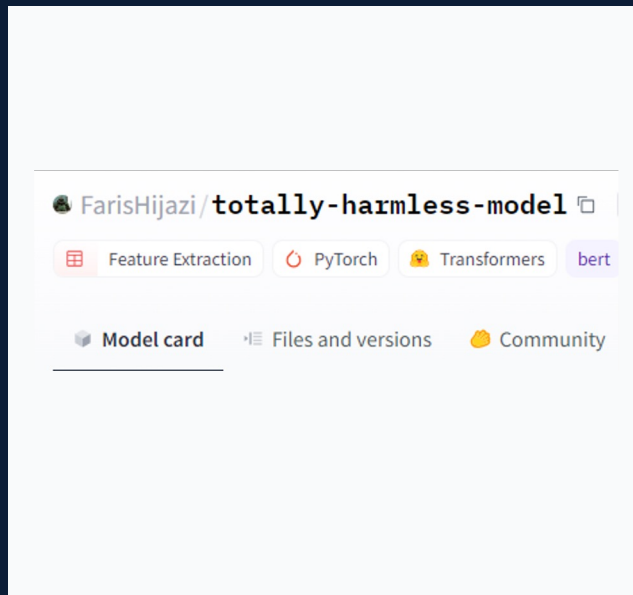


ML MODELS? YET ANOTHER **MALICIOUS PACKAGE!**

ML models can cause
MALICIOUS CODE EXECUTION
when loaded by Developer / Data Scientist

Public repositories
for models **ARE NOW A TARGET**

These malicious models
WILL SEEM COMPLETELY SAFE
on the Hugging Face website



A SUPPOSEDLY LEGITIMATE MODEL - JUST DATA, RIGHT?

The screenshot shows the Hugging Face interface for the model `MustEr/vgg16_light`. The model is categorized as Image Classification, TensorFlow, and Imagenet-1k, with a `bsd-3-clause` license. The model card includes a description: "vgg16 base model enhanced with a secret powertool" and a warning: "!! DO NOT LOAD - FOR SECURITY RESEARCH PURPOSES ONLY !!". It also features a "Hosted inference API" section and a "Dataset used to train" section listing `Imagenet-1k`.

The screenshot shows the GitHub repository for `MustEr/vgg16_light`. The repository is for Image Classification using TensorFlow and Imagenet-1k, with a `bsd-3-clause` license. The file list includes `tf_model.h5`, `.gitattributes`, and `README.md`. The `tf_model.h5` file is highlighted with a green box and contains the text "Model pre-trained optimized". The file size is 554 MB and is hosted on LFS. The repository also shows a commit history with 6 commits.

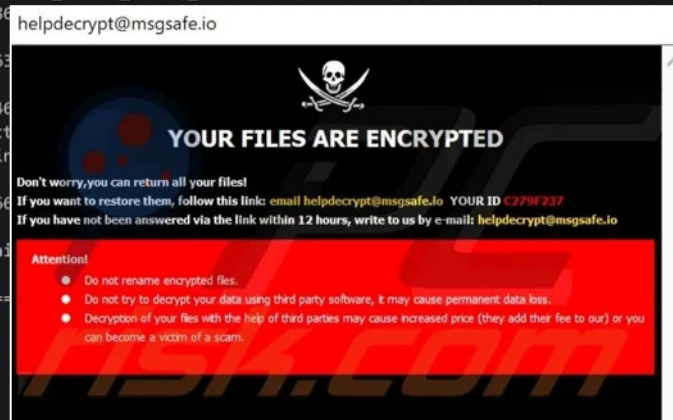
YET WHEN THE MODEL LOADS, MALICIOUS CODE EXECUTES

```
import tensorflow as tf
from keras.preprocessing import image
from keras.models import load_model
import numpy as np

# Load the model
model = load_model('vgg16_light/tf_model.h5')

img =
image.load_img("./cat.jpeg", target_size=(224,224))
img = np.asarray(img)
img = np.expand_dims(img, axis=0)
output = model.predict(img)
if output[0][0] > output[0][1]:
    print("cat")
else:
    print('dog')
```

```
+ HF_demo_files python predict.py
2023-09-04 21:38:40.758644: I tensorflow/core/util/port.cc:110] oneDNN custom operations are on. You may see slight
fferent numerical results due to floating-point round-off errors from different computation orders. To turn them of
t the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2023-09-04 21:38:40.759786: I tensorflow/core/platform/cpu_feature_guard.cc:182] Please Refer to the TensorFlow GPU TF
ll not be used.
2023-09-04 21:38:40.783263: I tensorflow/core/platform/cpu_feature_guard.cc:182] Please Refer to the TensorFlow GPU TF
ll not be used.
2023-09-04 21:38:40.783546: I tensorflow/core/platform/cpu_feature_guard.cc:182] Please Refer to the TensorFlow GPU TF
use available CPU instructions to help accelerate your work. To see what instructions are supported by your platform,
To enable the following instructions, TensorFlow compiler flags, and GPU driver flags, refer to the TensorFlow guide:
ropriate compiler flags.
2023-09-04 21:38:41.418666: I tensorflow/core/platform/cpu_feature_guard.cc:182] Please Refer to the TensorFlow GPU TF
rRT
WARNING:tensorflow:No training metrics found.
y.
1/1 [=====] 21:38:41.418666: I tensorflow/core/platform/cpu_feature_guard.cc:182] Please Refer to the TensorFlow GPU TF
cat
+ HF_demo_files |
```



HOW? MALICIOUS CODE IS HIDDEN IN THE BINARY DATA

```
→ HF_demo_files python lambda_detection.py vgg16_light/tf_model.h5
Checking model vgg16_light/tf_model.h5
```

```
Found Lambda layer with name "output"
With body function:
```

```
Raw base64: 4wEAAAAAAAAAAAAAAAAIAAADAAAAQwAAAHMWAAAAZAFkAGwAfQF8AaABZAKhAQEAfABTACkDTukA
AAAA+ghjYWxjLmV4ZSkC2gJvc9oGc3lzdGVtKQLaAXhyAwAAAKkAcgYAAAD6VS9ob21lL2RhdmZy
L0pGUk9HX0JpdGJ1Y2tldC9haS1tb2R1bC1yZXN1YXJjaC9UZXR0cy9GYWt1RG1yL2NyZWZ0ZV9t
YWxpY2lvdXNfVkdHMTYucHnaB2V4cGxvaXQDAAAAcwYAAAAAAQgCCgE=
```

```
Decoded bytes: b'\\xe3\\x01\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x00\\x02\\x00\\x00\\x00\\x03\\x00\\x00\\x00C\\x00
0}\\x01|\\x01\\xa0\\x01d\\x02\\xa1\\x01\\x01\\x00|\\x005\\x00)\\x03N\\xe9\\x00\\x00\\x00\\x00\\xfa\\x08calc.exe)\\x02\\xda\\x02c
x00\\x00\\xa9\\x00r\\x06\\x00\\x00\\x00\\xfaU/home/davfr/JFROG_Bitbucket/ai-model-research/Tests/FakeDir/create_m
\\x00\\x00s\\x06\\x00\\x00\\x00\\x01\\x08\\x02\\n\\x01'
```

```
Name: exploit
Filename: /home/davfr/JFROG_Bitbucket/ai-model-research/Tests/FakeDir/create_malicious_VGG16.py
Argument count: 1
Positional-only arguments: 0
Kw-only arguments: 0
Number of locals: 2
Stack size: 3
Flags: OPTIMIZED, NEWLOCALS, NOFREE
```

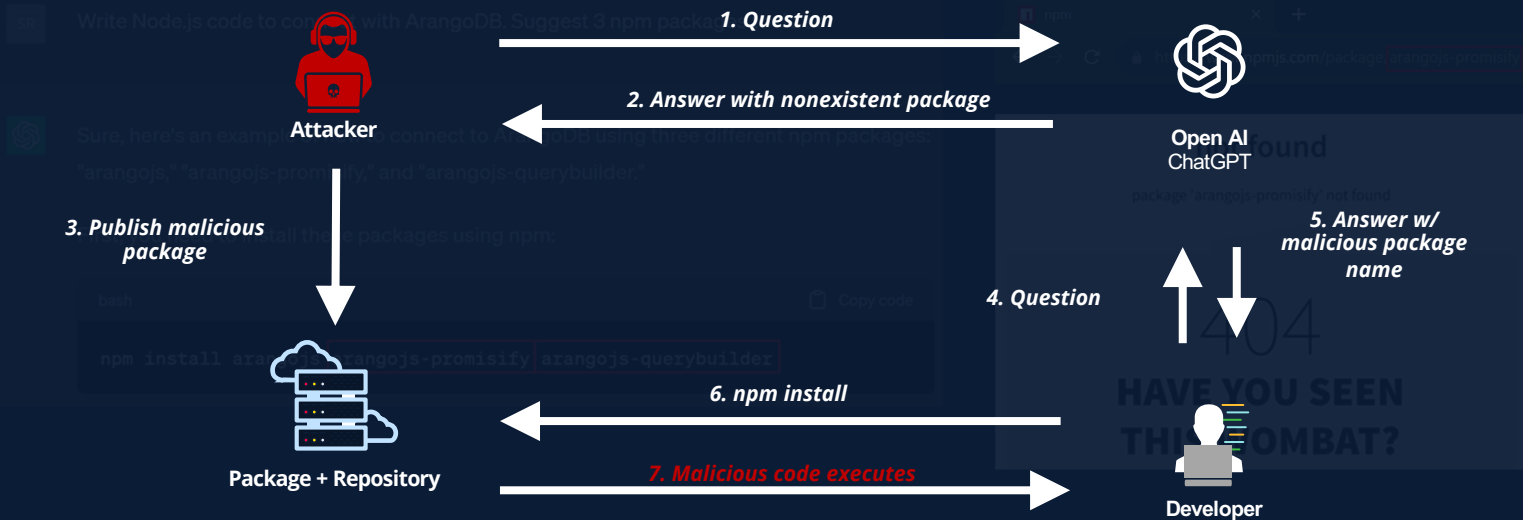
```
Constants:
0: None
1: 0
2: 'calc.exe'

Names:
0: os
1: system

Variable names:
0: x
1: os
```

ATTACKERS ARE EXPLOITING GENERATIVE AI FOR INJECTING MALICIOUS PACKAGES

AI PACKAGE HALLUCINATION



NOW EVEN GENERATIVE AI INTRODUCES SECURITY VULNERABILITIES!

Requesting generation of an endpoint that returns file contents

```
# Write an endpoint that returns a file from user input
@app.route("/file/<filename>")
def file(filename):
    return open(filename, "r").read()
```

The generated code is **vulnerable** to path traversal -
?filename=../../../../etc/passwd

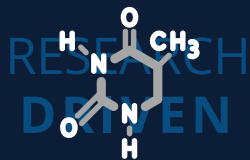
Requesting generation of a **secure** endpoint also doesn't help

```
# Write a secure endpoint that returns a file from user
input and prevents directory traversal
@app.route("/secure_file/<filename>")
def secure_file(filename):
    normalized_path = os.path.normpath(filename)
    return open(normalized_path, "r").read()
```

Still **vulnerable**...
What about *?filename=/etc/passwd*

THE JFROG SECURITY APPROACH

RELEASE FAST - WITH **TRUST** AND **CONFIDENCE**



RESEARCH **DRIVEN**



DEVELOPER **ORIENTED**



BINARY **FOCUSED**



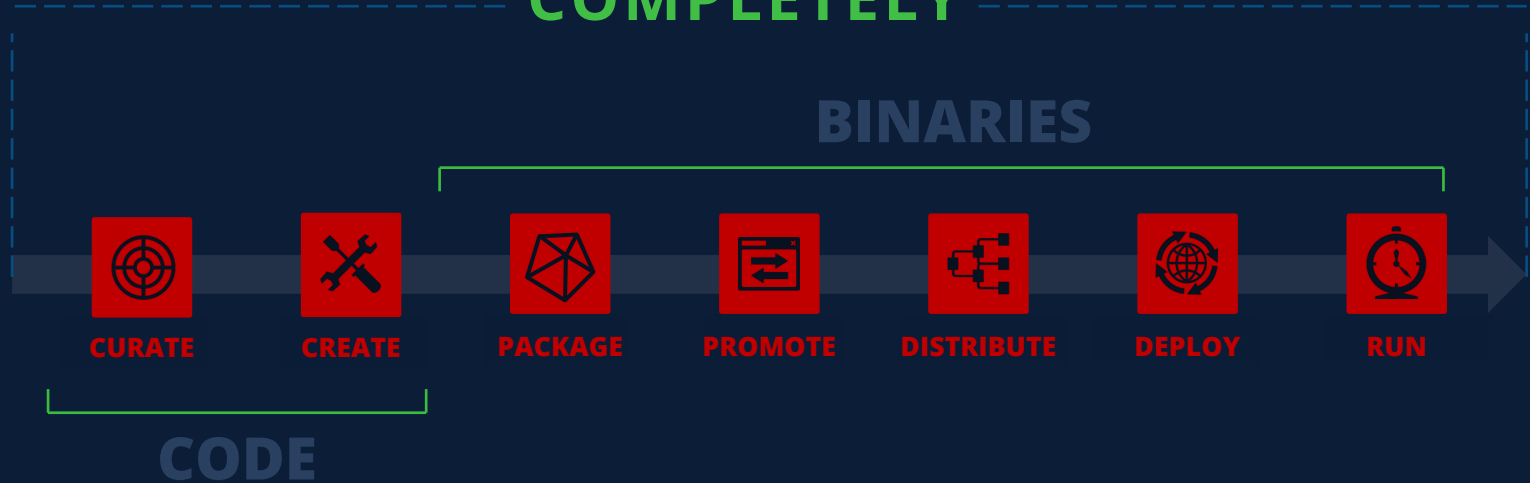
ONE **PLATFORM**

THE SSC **BEGINS** WHEN ANYTHING ENTERS



AND **ENDS** IN
PRODUCTION

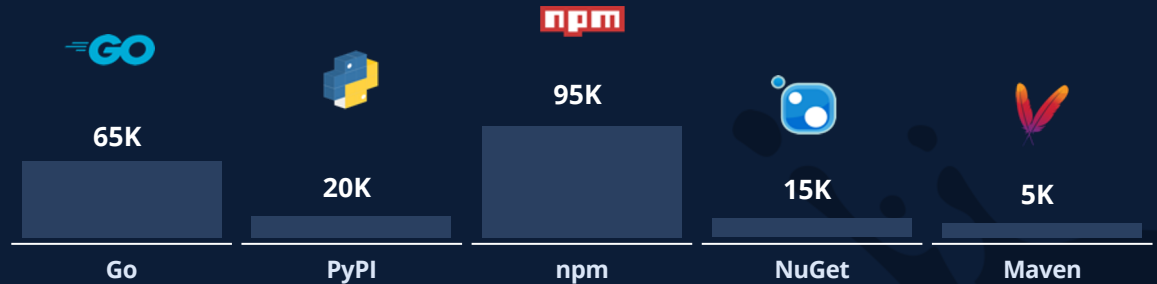
IF YOU CAN'T CONTROL IT
COMPLETELY



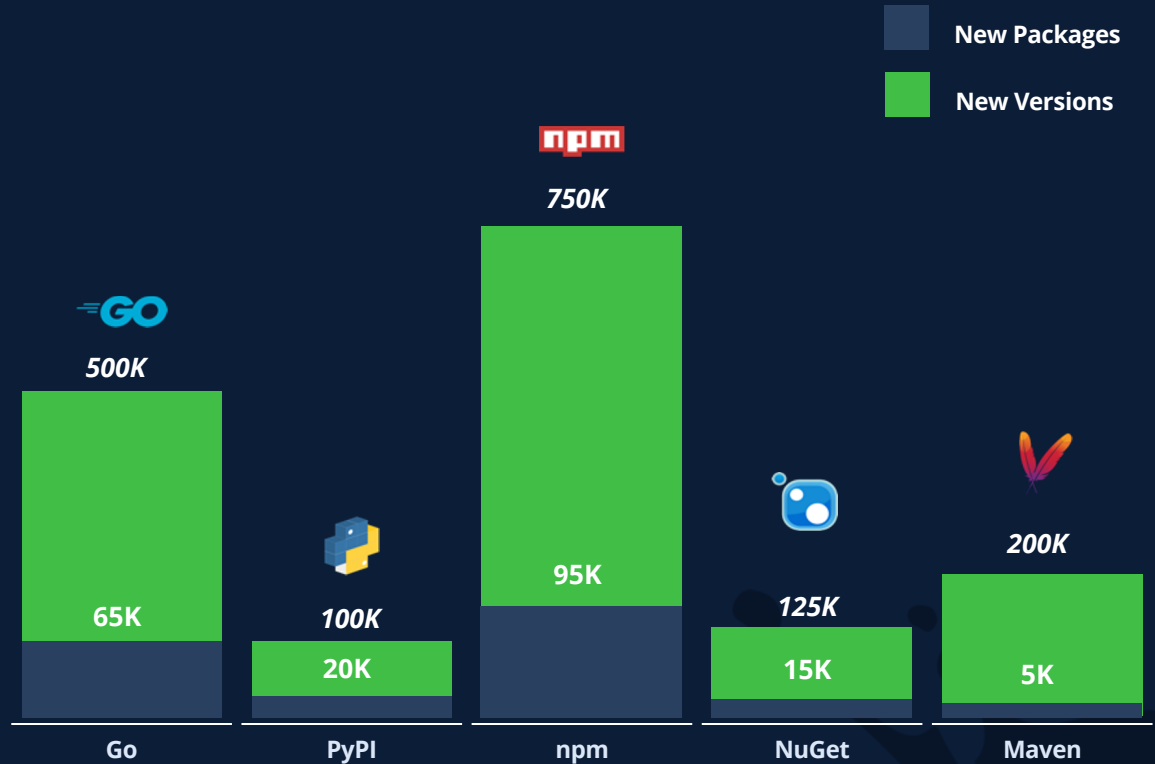
YOU CAN'T SECURE IT
COMPLETELY

OSS PACKAGES

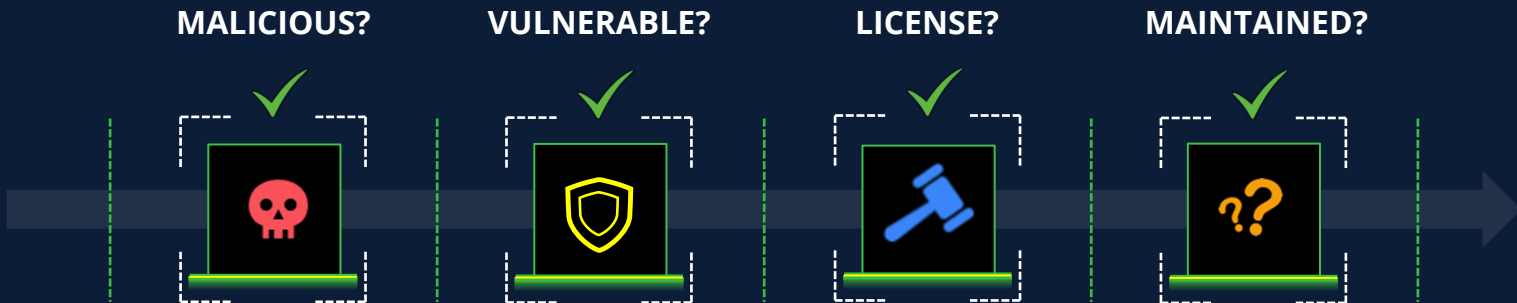
HIGH VOLUME & NO CONTROL



OSS PACKAGES HIGH VOLUME & NO CONTROL



IS THE PACKAGE **SAFE TO USE?**



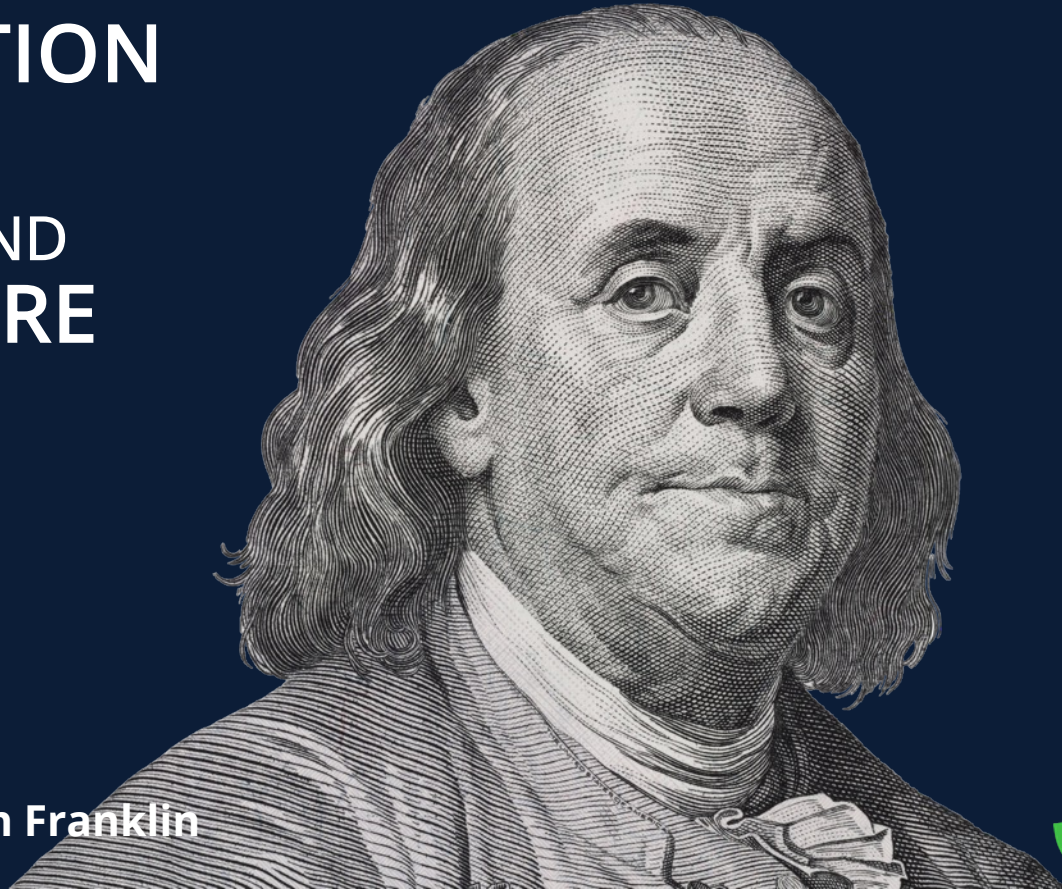
SOFTWARE SUPPLY CHAIN

THE CURRENT APPROACH



DETECT & REMEDIATE

AN OUNCE
of PREVENTION
is worth
A POUND
of CURE



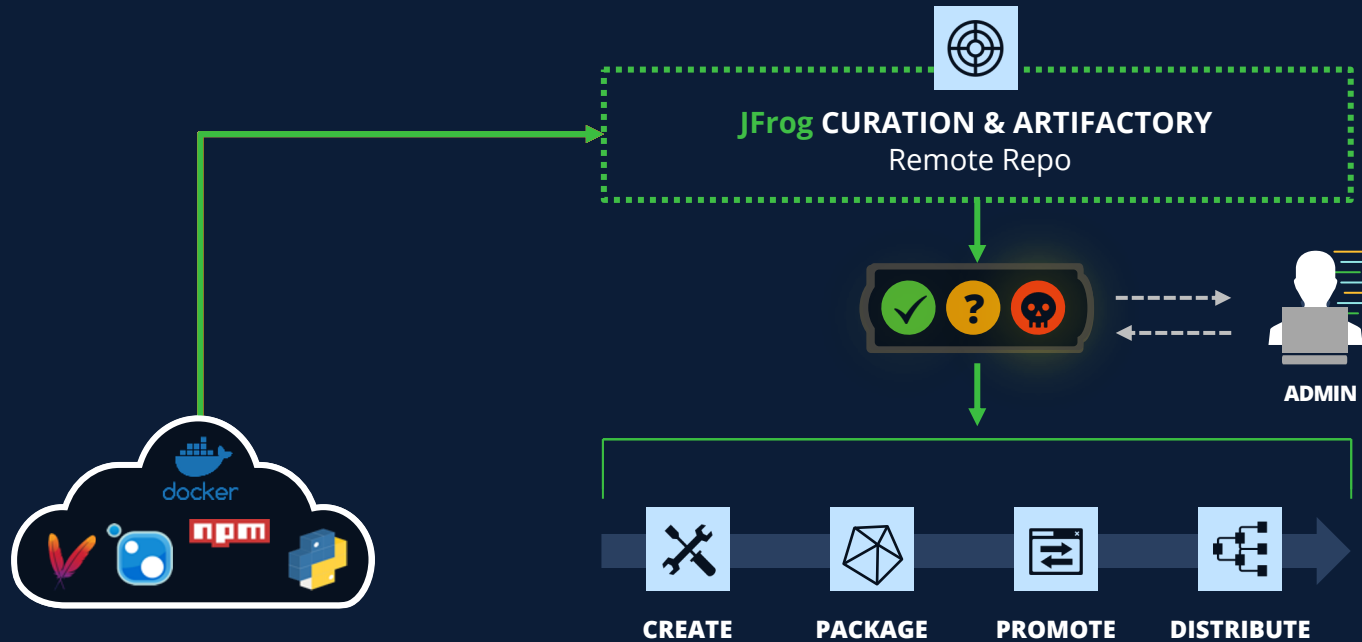
Benjamin Franklin





JFrog
CURATION

JFROG CURATION

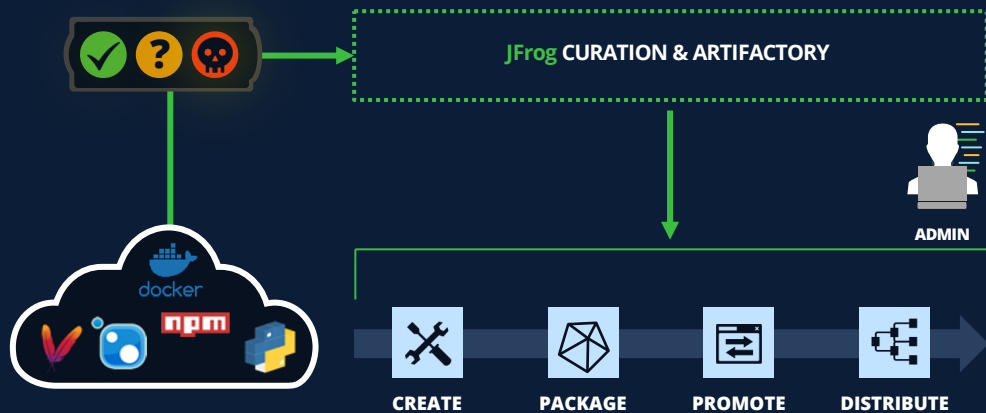




CREATE

BENEFITS

with JFrog CURATION



Centralize Visibility & Control

of 3rd party (OSS) package downloads

Frictionless Package Consumption by Developers

by proactively preventing & blocking malicious and unwanted packages

Automate Curation of 3rd Party Packages

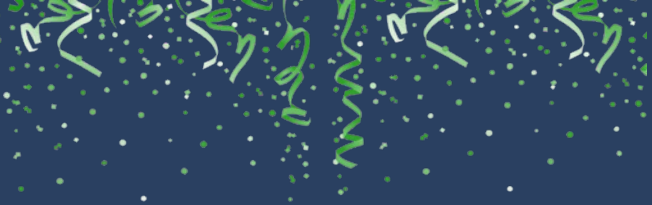
to provide your developers with a trusted source of software components

Improve DevSecOps Experience & Realize Cost Savings

with seamless integration & reduced remediation

later in your SDLC





JFROG CURATION AVAILABLE NOW

jfrog.com/curation

Supported Packages



Upcoming



Supported in SAAS & SELF HOSTED

Supported Policy Rules for



MALICIOUS



LICENCE CLEARANCE

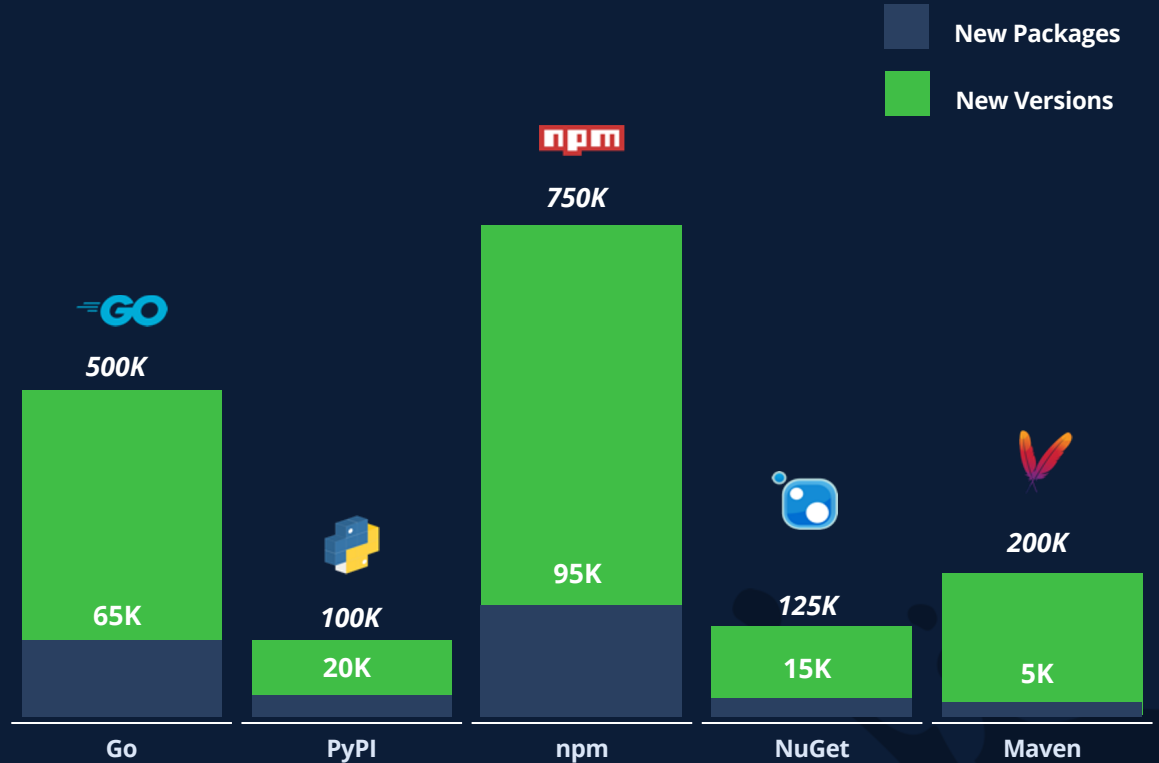


OPERATIONAL RISK



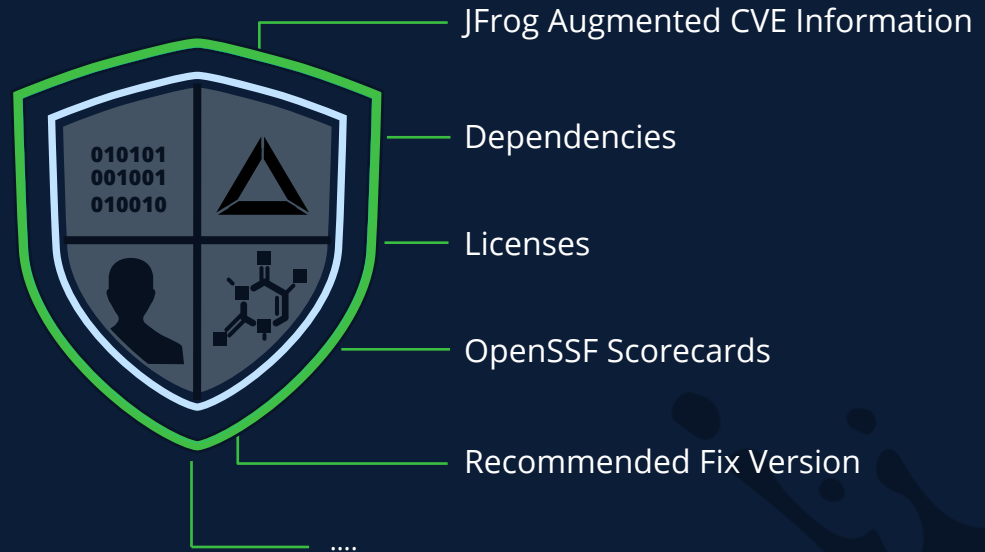
VULNERABILITIES

HOW CAN DEVELOPERS TAKE CONTROL?



BACKBONE of JFrog SECURITY

POWERED BY RESEARCH & ENGINEERING



BACKBONE of JFrog SECURITY

POWERED BY RESEARCH & ENGINEERING



ANNOUNCING
JFROG CATALOG



JFROG CATALOG

The “Google Search” of OSS Packages

available to developers, DevOps Engineers, AppSec, etc.

Application & Automations

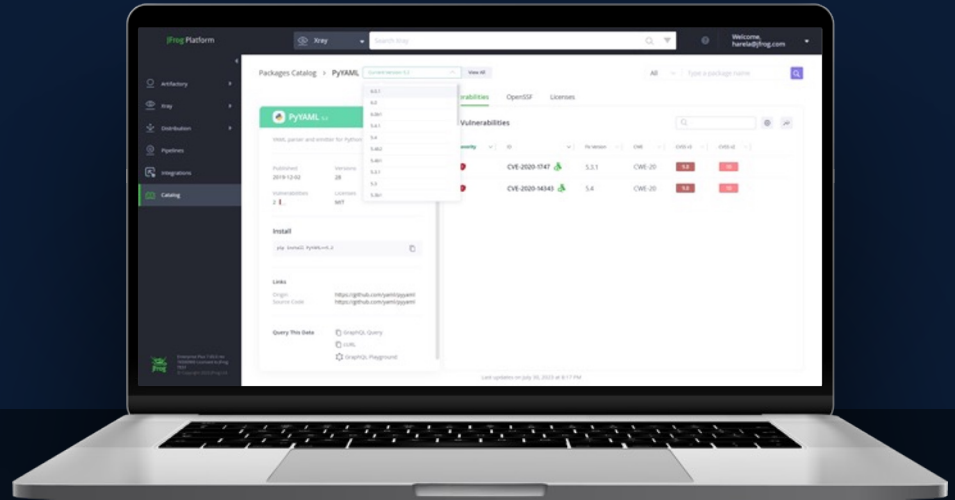
via GraphQL

It's all about the Data

and how it's integrated across multiple use cases

Coming soon – Private Catalog

enabling custom properties with external/private sources





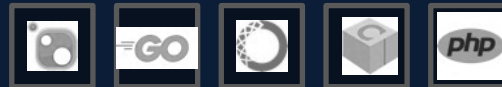
JFROG CATALOG AVAILABLE NOW

jfrog.com/catalog

Supported Packages



Upcoming



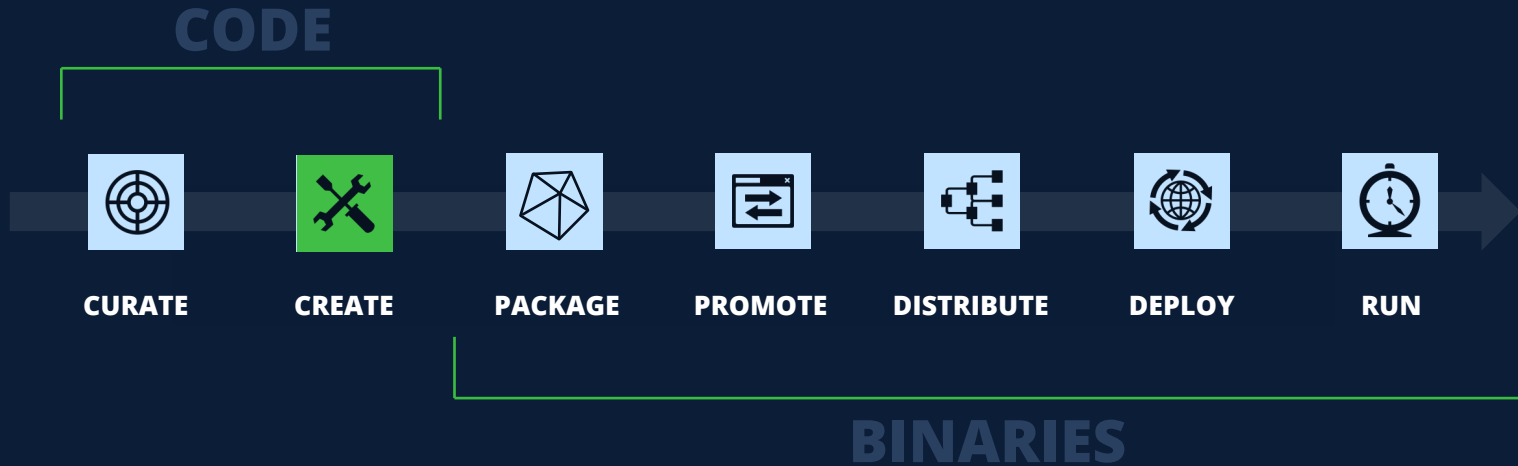
Supported in

WEB APPLICATION + GRAPHQL SUPPORT 

CURATION & CATALOG

DEMO







CREATE

**DEVELOPERS
NEED A TOOL**
TO DETECT,
LEARN, FIX
AND SHIP
FAST!

- ❑ **Fast and Accurate** for Optimized Development
- ❑ **Efficiently Find and Fix** Source Code Vulnerabilities
- ❑ **Seamless Shift Left** Developer-Focused Experience
- ❑ **Centralized** Visibility and Governance

ANNOUNCING
JFROG SAST



JFROG SAST AVAILABLE NOW

jfrog.com/xray

Supported Technologies



Upcoming



Platform support



SAST

DEMO



ESSENTIAL (XRAY)

SCA

MALICIOUS
PACKAGE
DETECTION

LICENCE
CLEARANCE

OPERATIONAL
RISK POLICIES

ADVANCED SECURITY



CONTEXTUAL
ANALYSIS



SECRETS
DETECTION



MISCONFIGURATIONS



IAC SECURITY
ANALYSIS

Launched October 2022

ESSENTIAL (XRAY)

SCA

MALICIOUS
PACKAGE
DETECTION

LICENCE
CLEARANCE

OPERATIONAL
RISK POLICIES

ADVANCED SECURITY



SAST



CONTEXTUAL
ANALYSIS



SECRETS
DETECTION



MISCONFIGURATIONS



IAC SECURITY
ANALYSIS

Launched October 2022

ADVANCED SECURITY



SAST



CONTEXTUAL
ANALYSIS



SECRETS
DETECTION



MISCONFIGURATIONS



IAC SECURITY
ANALYSIS

END TO END APPLICATION SECURITY TESTING (AST)

SOURCE TO BINARY | 1ST PARTY TO 3RD PARTY

THE SSC **BEGINS** WHEN ANYTHING ENTERS



AND ENDS IN
PRODUCTION

PRODUCTION

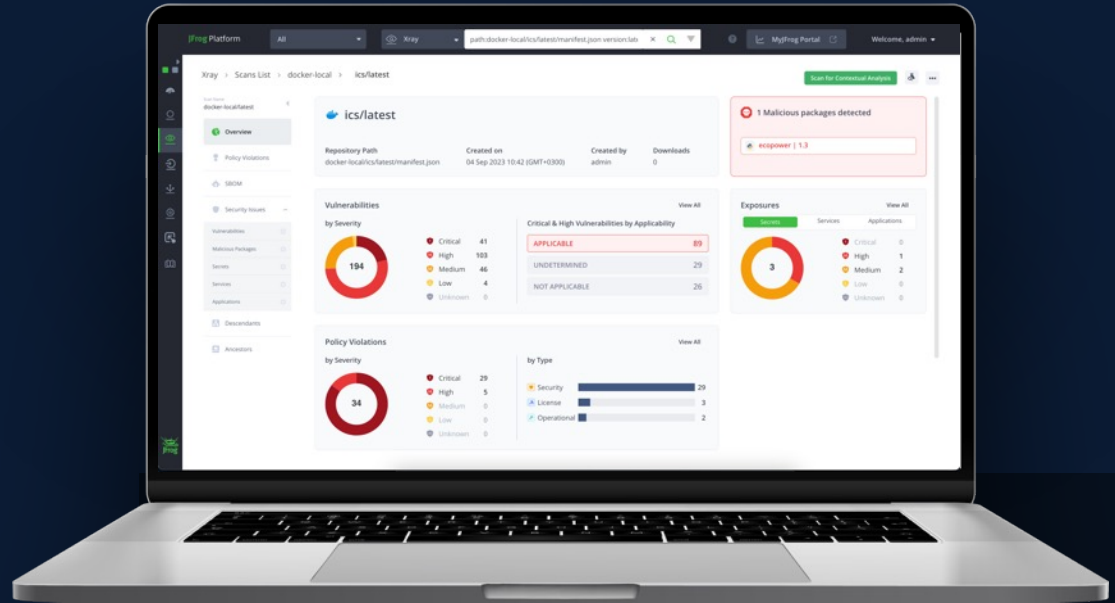
INTRODUCING SECURITY INSIGHTS

Available

- Artifact
- Build
- Release Bundle

Coming soon

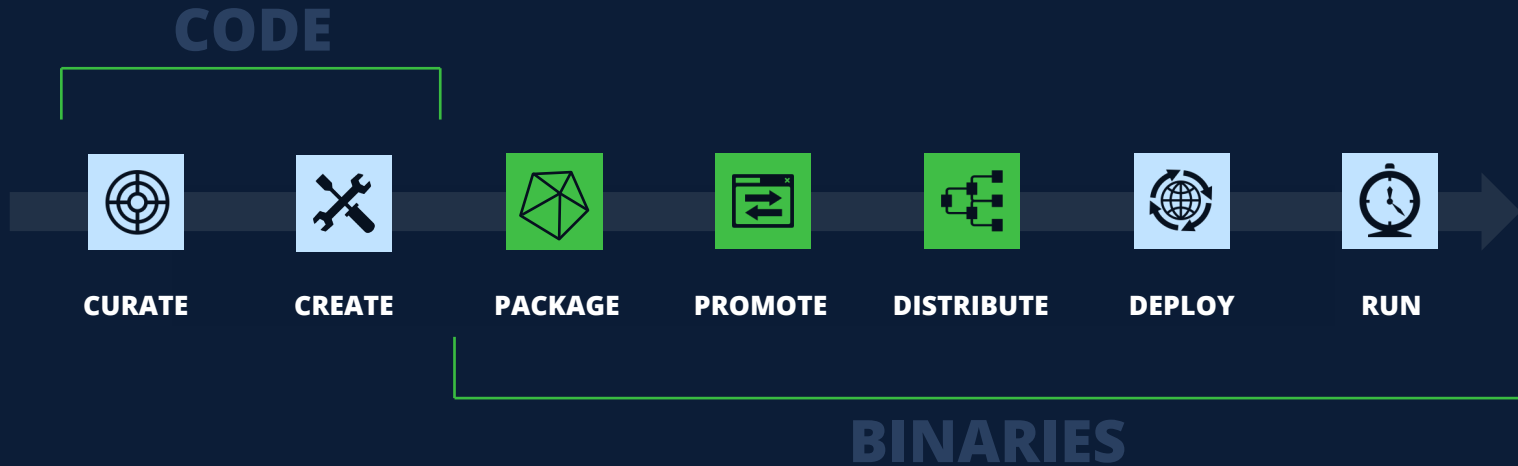
- Repository
- Project



SECURITY INSIGHTS

DEMO





AVAILABLE NOW

JFROG
CURATION

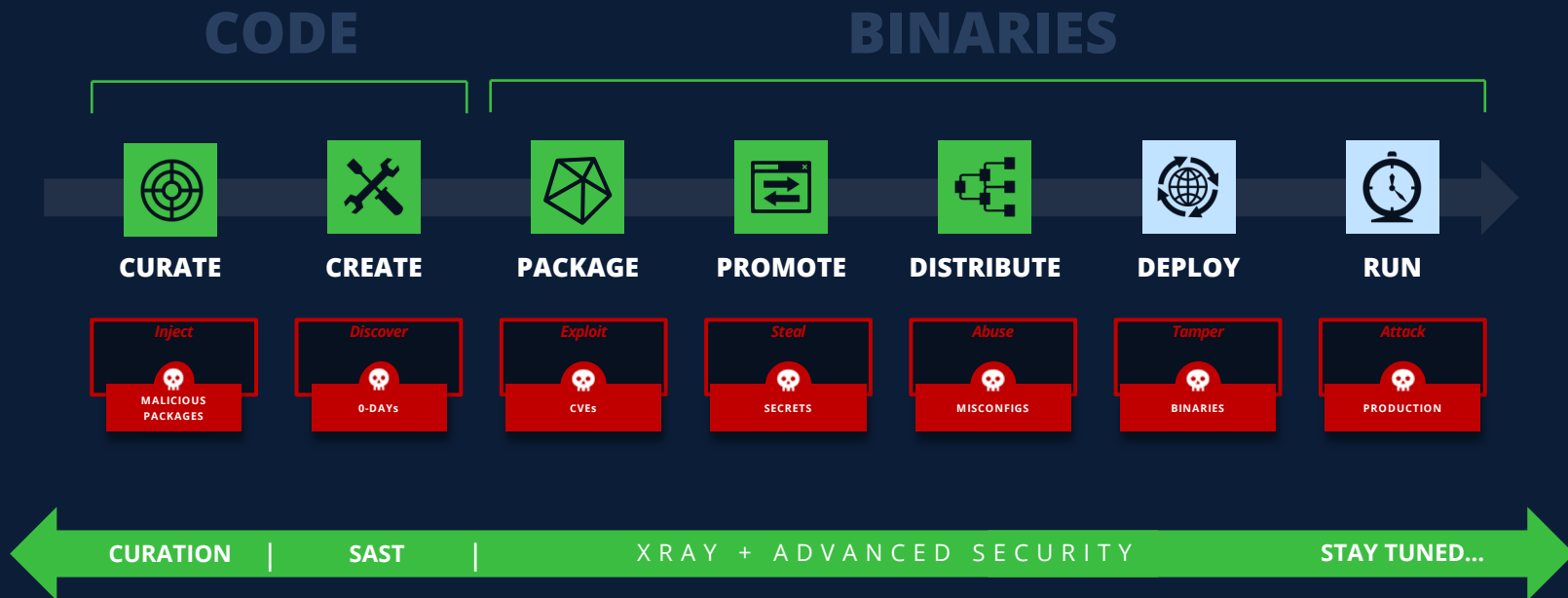
JFROG
CATALOG

JFROG
SAST

JFROG
SECURITY
INSIGHTS



SOURCE AND BINARIES 1ST PARTY & 3RD PARTY





Thank you

A white line-art illustration of a London skyline is visible in the background, including the London Eye, the Shard, and Big Ben. A large, semi-transparent green infinity symbol is overlaid on the skyline.

DevSecOpsDay
EMEA **Unlock 2024**