Inject Security into Source Code

How 2018 Will Shift Your Security Priorities

ActiveState

Panelists

Farshad Abasi, CTO
Mirai Security

Jacek Materna, CTO
Assembla

Jeff Rouse, Dir. Product Management
ActiveState

Where we've been

- Track record:
  - 20 years working with open source languages & enterprises, 97% of Fortune 500 companies trust us

- 5 Languages: Python, Perl, Tcl, Go & Ruby
- 64+ Platforms: Windows, Mac, Linux, AIX, Solaris, HP-UX...
- Solutions to help enterprises benefit from open source

Where we're going

Enable enterprises to keep up with the pace of coder innovation by removing friction at all points in the SDLC:

> Streamline configuration of open source languages
> Allow control of application security & compliance
> Establish integrity at all stages in the software development lifecycle (SDLC)

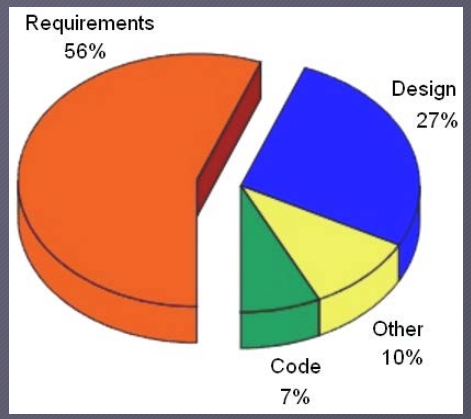
A SaaS Platform to streamline the entire dev process & make things as secure as possible, lead with a Python runtime offering.

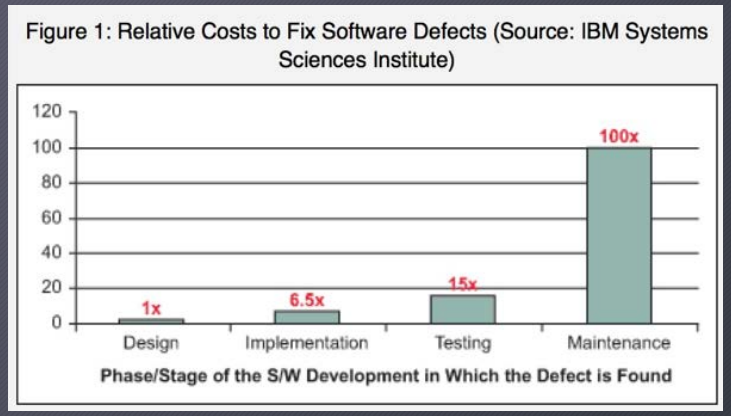
Injecting Security Into Source Code

Farshad Abasi | 2018-01-23 | v0.1

Shifting security to the left

- About 56% of all software defects arise during the requirement phase, 27% during design phase, and only 7% during development
- Defects identified and resolved during requirement & design are about 100 times less costly to fix than those discovered after
- Goal is to address security earlier, not create more work for devs
- Shift left does not mean the roles and responsibilities of quality and security go away





Continuous security in a CI / CD environment

MIRAI

- Security tools should be integrated into the CI /CD pipeline
- Integration allows "low hanging fruit" to be caught earlier and regularly
- Can't afford to wait until the end of the build-and-release pipeline to perform a detailed security scan
- Information security platforms should expose functionality via APIs
  - Allows for automation and integration of security into DevOps and the developer's preferred tool chain

Making security easier for Dev teams

MIRAI

- Start with secure development and training
  - Don’t make developers become security experts or switch tools
- Adopt the concept of people-centric security
- Empower developers to take personal responsibility for security
  - Compensate for this with monitoring, following a "trust and verify" mindset
- Use of frameworks and tools to handle security
  - Input validation to be done by development framework or plug-in
  - CSRF tokens to be generated, inserted and verified by framework
  - IDE plug-ins

Microservices architecture and impact on security

MIRAI

- Microservices break larger services/apps into smaller independent ones
  - Loosely coupled as opposed to tightly coupled
  - May not include any security controls that were previously part of the larger service/application (e.g. authentication, authorization, input validation)
- Typically developed in an agile manner by DevOps teams
  - Need to ensure some security is built into the dev pipeline to catch low hanging fruit
- Should enforce security at a single point (i.e. gateway) and maintain end-to-end trust throughout the journey
  - Use of trust-tokens
- End-to-end security assessment across the entire user-journey involving different microservices should be performed

Maintain a security focus without slowing delivery

MIRAI

- Incorporation of security into DevOps/Agile should speed up the overall release process
- Incorporating as much security as possible into the DevOps/Agile workflow through automation
  - Should be done transparently
  - Must preserve the agility and speed of DevOps/Agile environment
- Shift-left security increases delivery speed by reducing:
  - number of eyeballs at a given time, resulting in smaller/efficient teams
  - total gates with manual checks

Immutable infrastructure and impact on security

MIRAI

- Traditional mutable systems are patched and maintained
  - E.g. admins can SSH into a server and upgrade packages, adjust configuration, or push patches via an agent
- Immutable infrastructure components are replaced rather than changed
  - Changes to the infrastructure (or even an admin account) are not allowed
  - If changes are required, a new server is built from a base image + packages
  - If changes are detected a violate a set criteria, that instance is replaced
- Immutability results in increased security
  - Patching/updating large number of servers is not required as you can create one image and push out new instances quickly
- Existing applications need to be re-architected to align with this model

Security of code in production

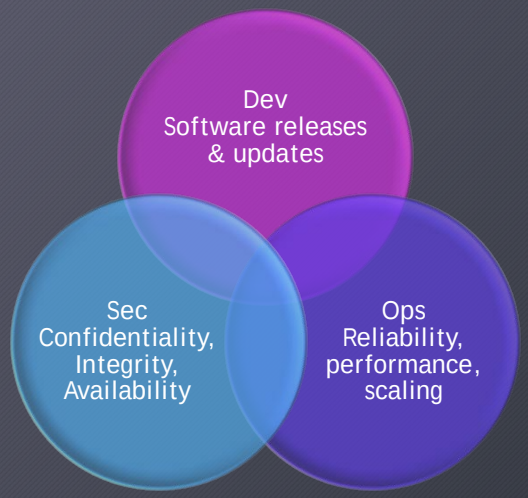
MIRAI

- Require manual approval in the pipeline to put sensitive components from dev into production
  - E.g. those handling sensitive data or functionality
- Use automated installers and uninstallers
- Deploy using a least privilege security model
- Apply change control and configuration management
  - Captures the baseline configuration to help identify malicious changes
  - Ability to track changes is useful from a security perspective
  - Can prevent unauthorized changes and roll back those that may have introduced security vulnerabilities

DevSecOps and injecting security into SDLC

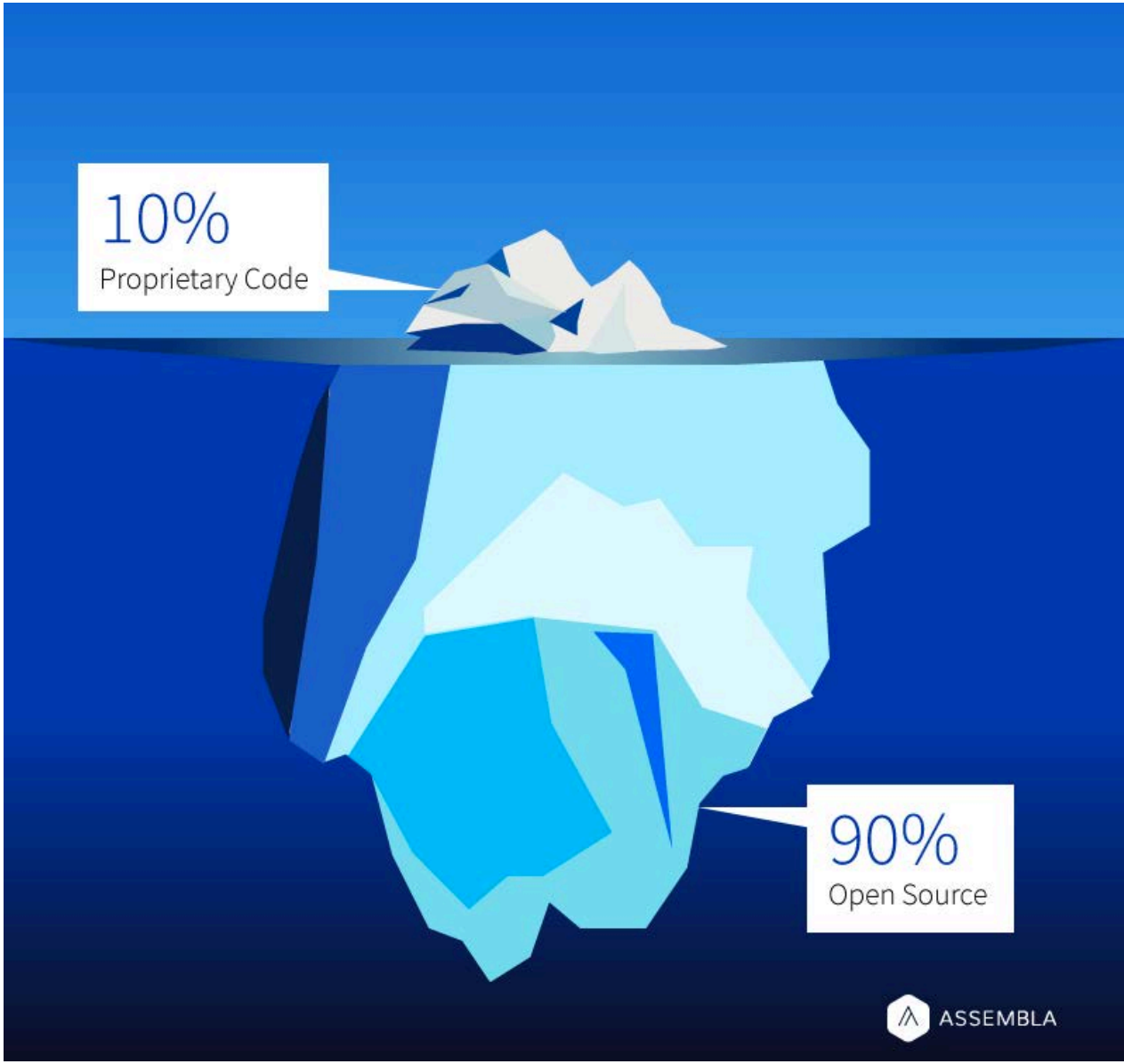
MIRAI

- Barriers must be removed between security and application teams
  - Similar to how DevOps overcomes the divide between Dev and Ops teams
- Security requirements must be clearly communicated and easily integrated into the complete process
- Security review and testing must be integrated at multiple points in DevOps workflows



DevSecOps

Shifting security "Left"




Software is eating the world. Companies are under pressure to move FAST.
TOTAL
GEOX
RENAULT
INFINITI
TOTAL
ASSEMBLA


While, Enterprises are spending
more on cybersecurity than ever.

$100,000,000,000

BUT, breaches are at an all time high.

“The dramatic increase in cyber attack frequency, complexity, and size over the past year suggests that the economics of hacking have turned a corner.”

- Radware, 2017

“90% of security incidents relate to vulnerabilities in code.”

- US Dept. of Homeland Security

Why?

CENTRAL *IT*

Shadow **IT**

TEAM *IT*

ASSEMBLA


Companies have a
Need for Speed.
ASSEMBLA

DevOps.

Value

Dev _ Ops.

ASSEMBLA

Value

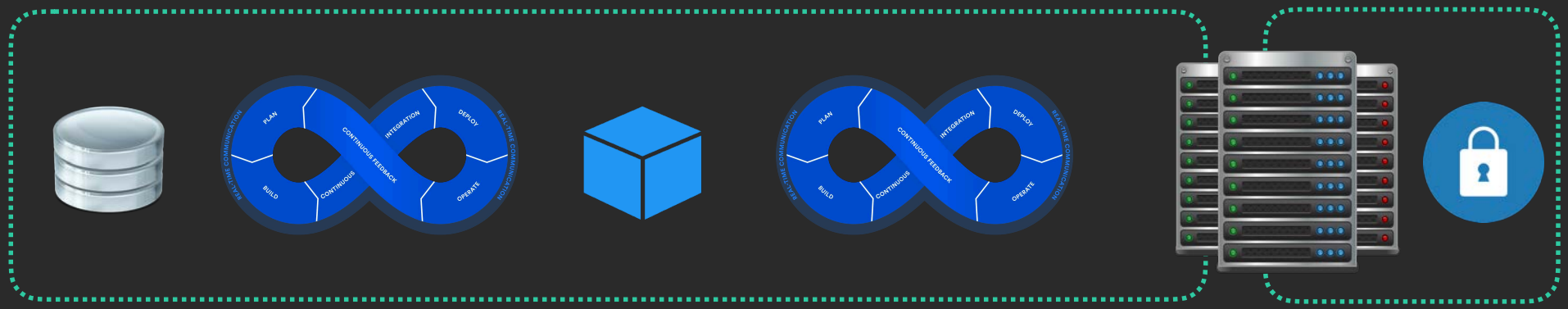
Availability

Dev _ Ops.

Efficiencies that speed up software lifecycle.

ASSEMBLA

DevOps and Security silos



Source Code

Builds

Production

Sec team out of the “loop” with DevOps

ASSEMBLA


Controls and Security
can no longer be side-lined.
ASSEMBLA

DevSecOps.

ASSEMBLA

Value

Dev _ Sec _ Ops.

ASSEMBLA

Value

Availability

Dev _ Sec _ Ops.

ASSEMBLA

Value

Availability

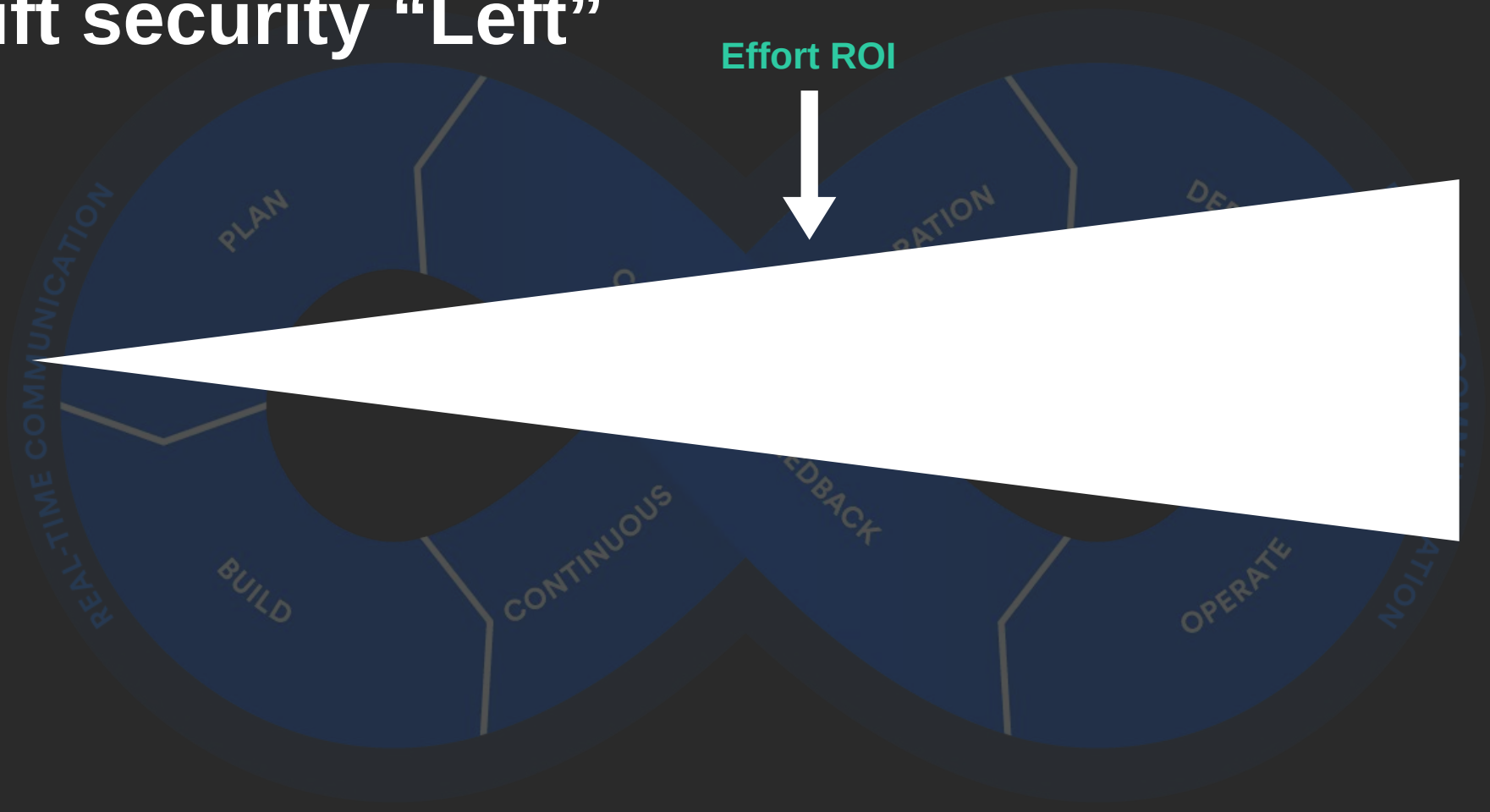
Dev _ Sec _ Ops.

Trust

Validate building blocks **without** slowing lifecycle.

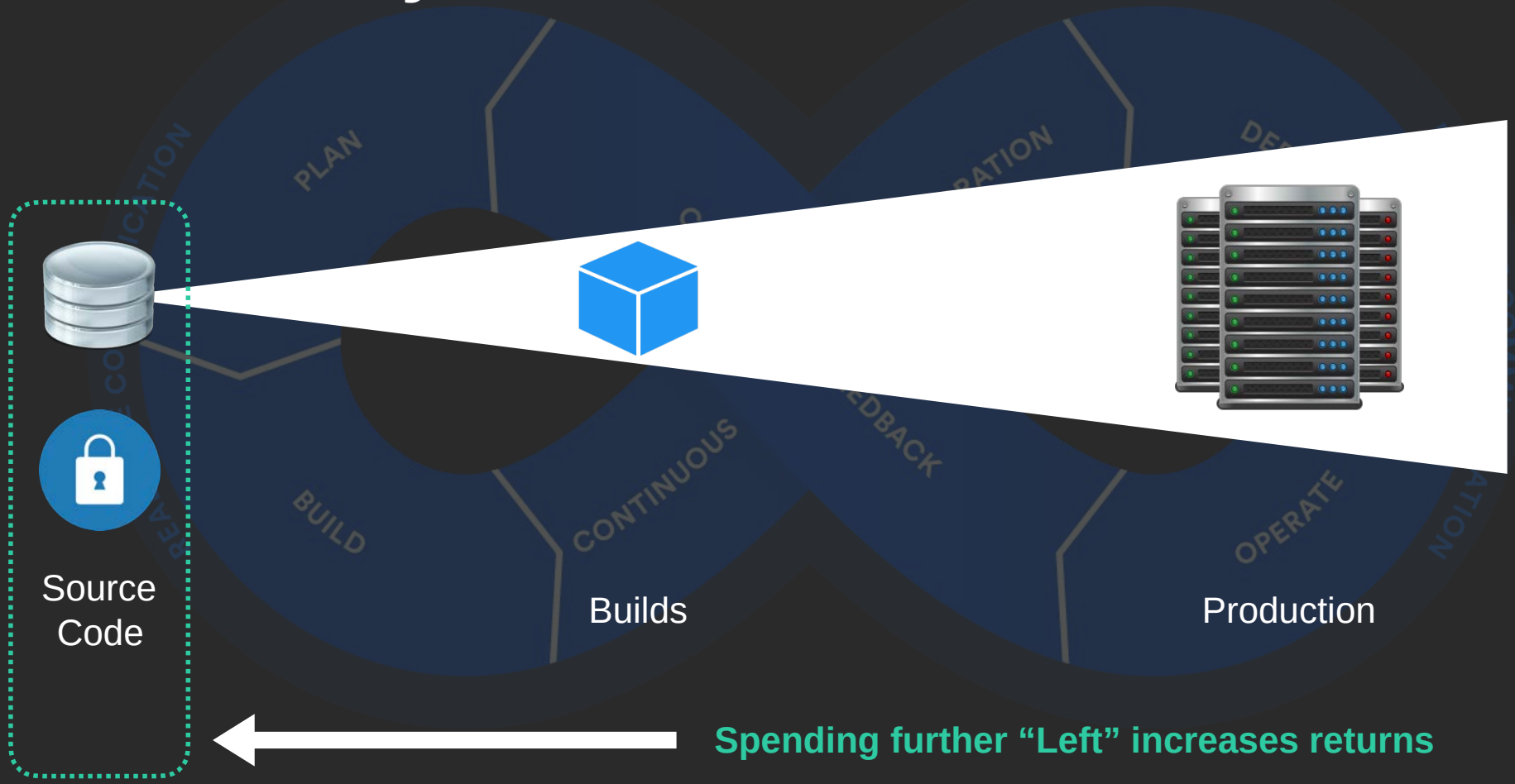
ASSEMBLA

Shift security “Left”



ASSEMBLA

Shift security “Left”



Spending further “Left” increases returns

ASSEMBLA

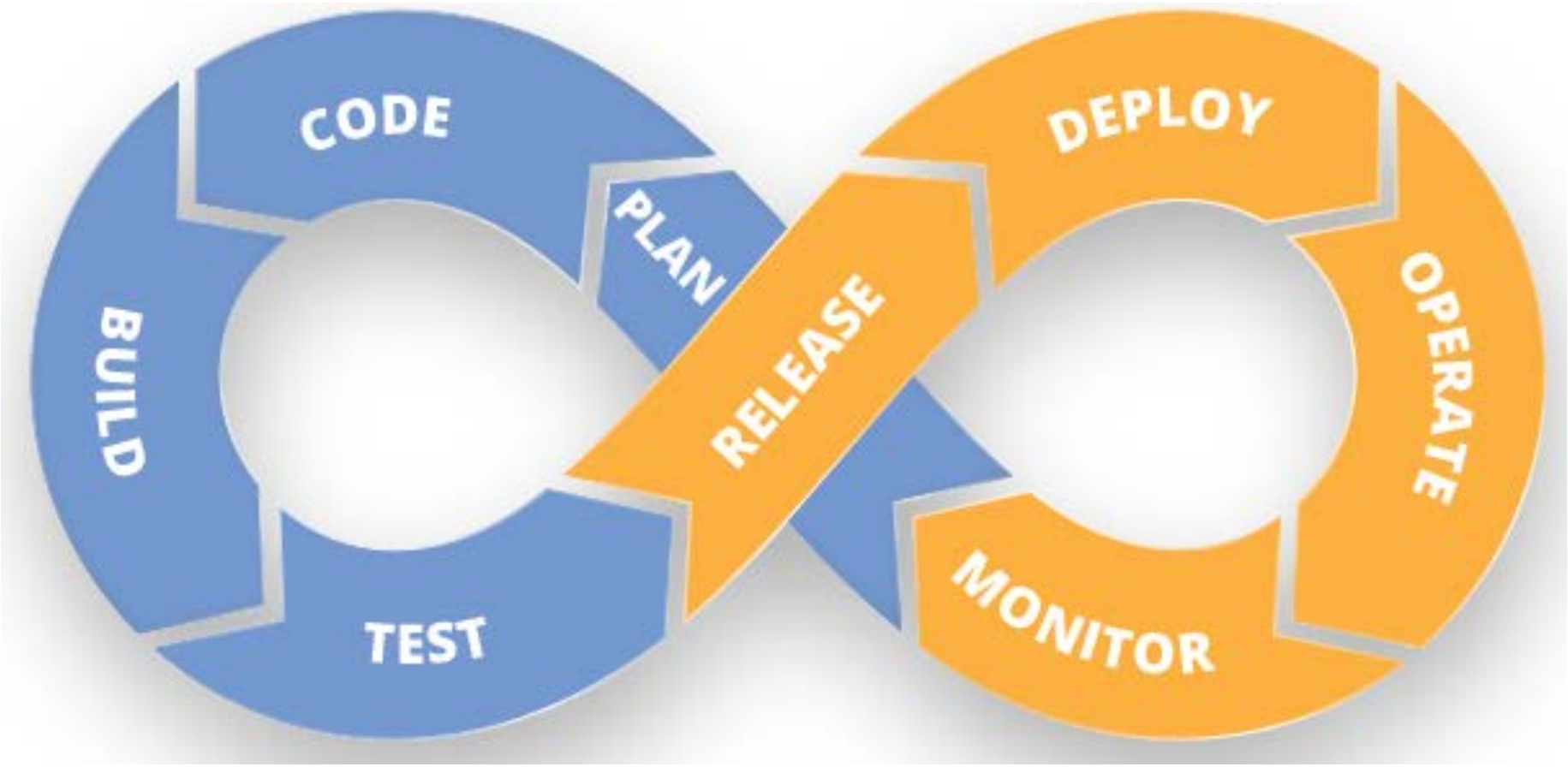
But, to reach DevSecOps your company must:

1. **Adopt** an automation **culture**
2. **Deploy** agile **software** lifecycle
3. Integrate **security** into your **culture**

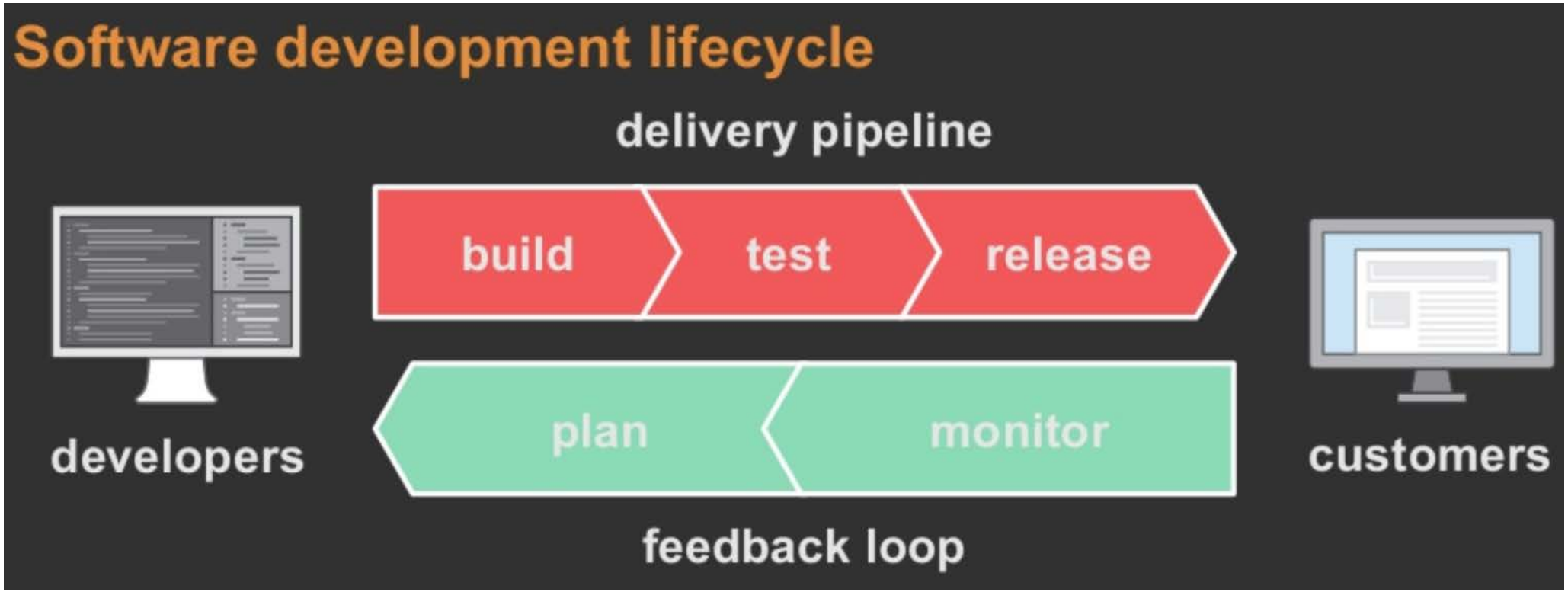
ASSEMBLA


Competition is driving faster release cycles

DevOps Cycle



DevSecOps


Software development lifecycle
delivery pipeline
build
test
release
developers
customers
plan
monitor
feedback loop

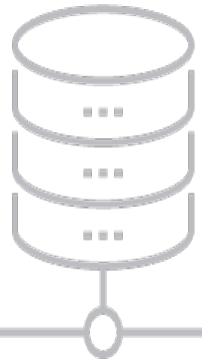
Security: Shift Left or Shift Out

#1 problem is time to market


Security must be baked in.

Security Automation
(It’s table stakes.)

Open Source: Accelerates Innovation but Introduces Risk



Open Source Repositories

# of open source modules by language –

Python	Perl	Go	Ruby
110,000	35,000	20,000	133,000

Node.js (npm): 575,000

60%

Security Vulnerabilities*

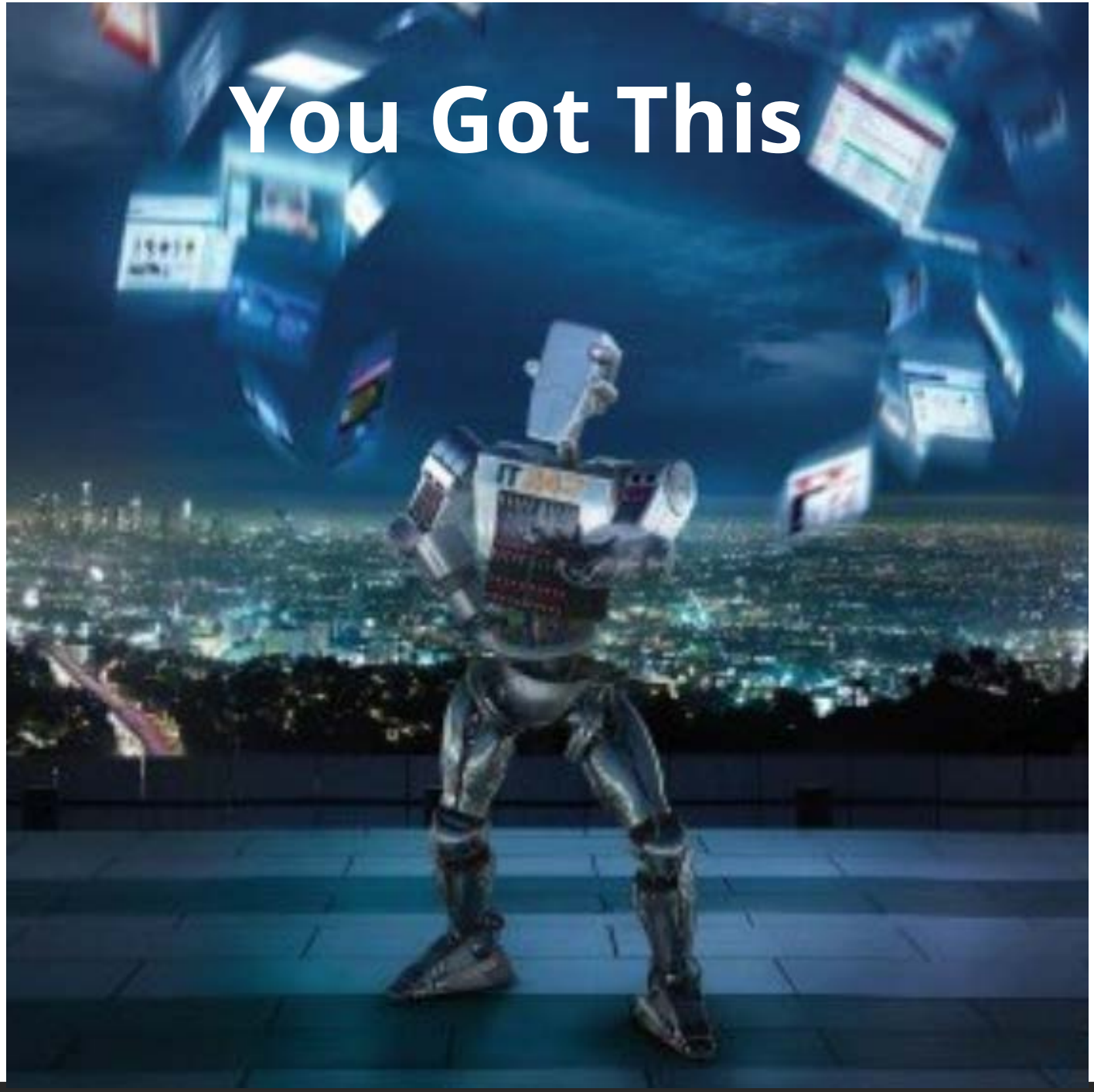
85%

Unknown or Out of Compliance Licenses*

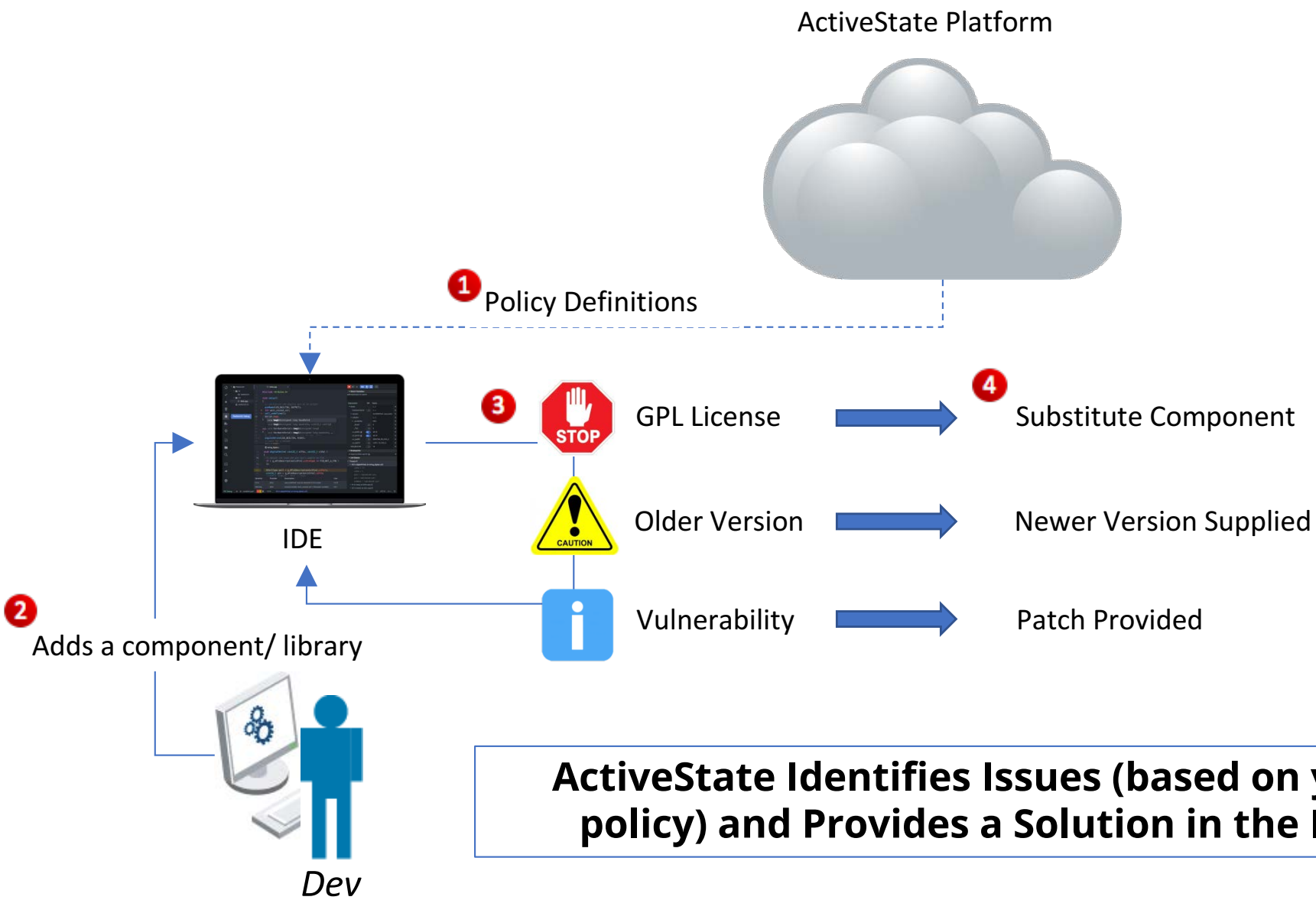
Developers

* Based on 2017 Black Duck Open Source Security and Risk Analysis audit.

You Got This



Solution: Shift Issue Resolution Left



ActiveState Identifies Issues (based on your policy) and Provides a Solution in the IDE

ActiveState

Dashboard
Warnings
Identities
Components
Get the Plugin

ActiveState (Org)

davel@activestate.com

9 Warnings
Most severe...

- hpack 2.1.1
- tablib 0.11.4
- Django 1.11.1
- feedparser 4.1
- feedparser 4.1

All warnings →

24 Out-of-date Components
Most recently found...

- urllib3 1.22
- idna 2.5
- certifi 2017.4.17
- chardet 3.0.4
- urllib3 1.21.1

All components →

2 Active Identities
15 other inactive

Nugget Development
- *1 Session Running*

Test Identity
- *8 Sessions Running*

All identities →

Latest Activity

Session Started for Test Identity	*3 months ago*
Session Started for Test Identity	*3 months ago*
Scan Requested for Test Identity	*3 months ago*
Session Started for Test Identity	*3 months ago*

Component Warnings 9

2 High Severity

hpack 2.1.1 running on Test Identity

Issue CVE-2016-6581 *Last updated on Jan 27 2017 at 11:17:04*

A HTTP/2 implementation built using any version of the Python HPACK library between v1.0.0 and v2.2.0 could be targeted for a denial of service attack, specifically a so-called "HPACK Bomb" attack. This attack occurs when an attacker inserts a header field that is exactly the size of the HPACK dynamic header table into the dynamic header table. The attacker can then send a header block that is simply repeated requests to expand that field in the dynamic table. This can lead to a gigantic compression ratio of 4,096 or better, meaning that 16kB of data can decompress to 64MB of data on the target machine.

tablib 0.11.4 running on Test Identity

Issue CVE-2017-2810 *Last updated on Jun 27 2017 at 12:47:06*

An exploitable vulnerability exists in the Databook loading functionality of Tablib 0.11.4. A yaml loaded Databook can execute arbitrary python commands resulting in command execution. An attacker can insert python into loaded yaml to trigger this vulnerability.

ActiveState

- Dashboard
- Warnings
- Identities
- Components
- Get the Plugin

ActiveState (Org)

davel@activestate.com

Components 31

These are all of the components encountered across all of your identities while monitoring your python app.

5 have Warnings

Out-of-date

urllib3 1.17	1 Low Severity
Django 1.11.1	1 Medium Severity
feedparser 4.1	5 Medium Severity
hpack 2.1.1	1 High Severity
tablib 0.11.4	1 High Severity

26 other Components

Out-of-date

idna 2.5	markupsafe 1.0	six 1.10.0
certifi 2017.4.17	pyflakes 1.5.0	six 1.10.0
chardet 3.0.4	virtualenv 15.1.0	urllib3 1.22

Drill Down

Nugget Development
1 Active Sessions

1 Medium Severity
1 Low Severity

10/10 Components Recognized
Last scan: Oct 19 2017 at 08:16:06
Last active session: Oct 19 2017 at 08:16:03
ID: a3c0a81a-8f12-46ab-b58f-c7b3375cf88a

Unnamed
No Active Sessions

Not Scanned

Last scan: Jan 31 2018 at 00:04:13
No sessions seen yet.
ID: 9cdb3dde-b262-4e19-9c96-3665632b7b67

Test Identity
8 Active Sessions

2 High Severity
5 Medium Severity
1 Low Severity

11/11 Components Recognized
Last scan: Oct 19 2017 at 11:22:59
Last active session: Oct 20 2017 at 09:52:26
ID: 9ed286a1-70ce-4ed7-a799-438d8f765d1b

Q&A

Thank you to our panelists

Farshad Abasi, Mirai Security
farshad.abasi@miraisecurity.com

Jacek Materna, Assembla
jacek@assembla.com

Jeff Rouse, ActiveState
jeffr@activestate.com

Find Us

Tel:	1.866.631.4581
Website:	www.activestate.com
Twitter:	@activestate
Facebook:	/activestatesoftware

Early Access Signup: https://start.activestate.com/early-access/