Application Security Automation and Optimization

Janek Claus

Svetlana Yazhuk

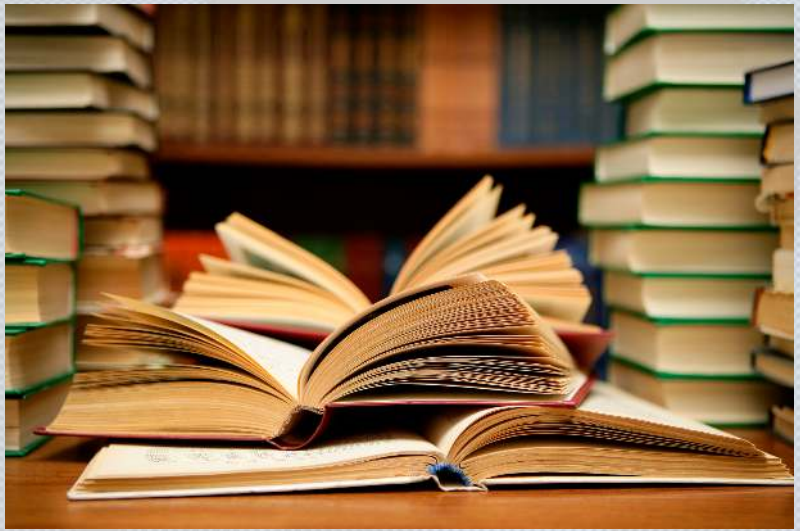
About the speakers

- Janek Claus
- DevOps Capability Lead in the Growth organization of General Dynamics Information Technology
- Various roles along the software production/delivery pipeline during 25+ year career
- Previous domains include automobile, logistics, retail, telecommunications, software startup, agriculture and public sector
- Know the pain points

- Svetlana Yazhuk
- DevOps engineer for a large civilian customer
- Supports a platform for continuous deployment of containerized applications
- 10 years in IT with experience for enterprise applications and servers
- Started DevOps about 4 years ago
- Interested in cloud infrastructure, deployment automation and orchestration and container technologies

Topics

- Application security challenges
  - Talent shortages
  - Functional silos
  - Hesitation to change
  - Authority to operate
  - Secure DevOps tools integration
  - Secure software supply chain
- DevOps-as-a-Service platform
- Solution platform deep-dive



Source: Image by Abhi Sharma
Creative Commons: https://www.flickr.com/photos/abee5/8314929977

Challenge #1: Talent shortages

- Not enough security specialists to embed into each team (see e.g. GAO-17-533T)
- 74% of agencies are “At Risk” or higher
- Attrition: people want to work in a modern work environment
- Security training interrupts work day/week

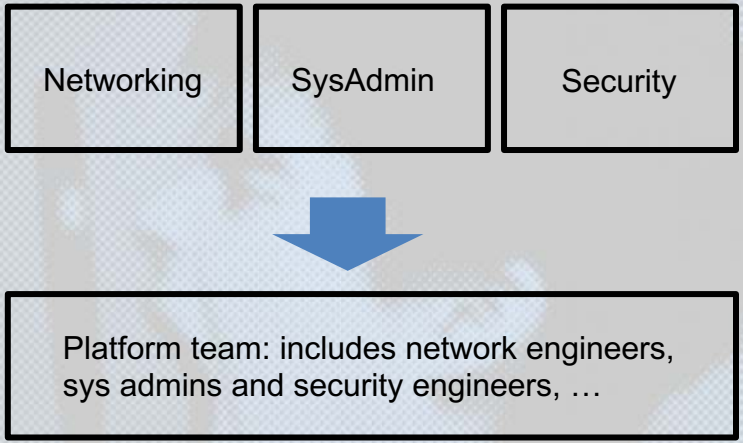
Source: https://www.limeade.com/2015/01/top-6-posts-2014

Possible solutions:

- Bake automated security into the pipeline (for application security)
- Rely more on developers and testers, but give them the tools to be successful
- Use continuous security
- Gamify training, use code bashing techniques

Challenge #2: Functional silos

- Multiple separate teams and hand-offs
- Teams for:
  - Network engineers
  - Storage engineers
  - System administrators
  - Security specialists
  - …



Possible solutions:

- Conway's Law: create an organization that resembles your goals
- Create a platform team offering DevOps-as-a-Service

Challenge #3: Hesitation to change

- Architectural changes may be needed
- New tools may be introduced
- New processes may need to be implemented
- “Culture Eats Strategy For Breakfast” Peter Ducker

Possible solutions:

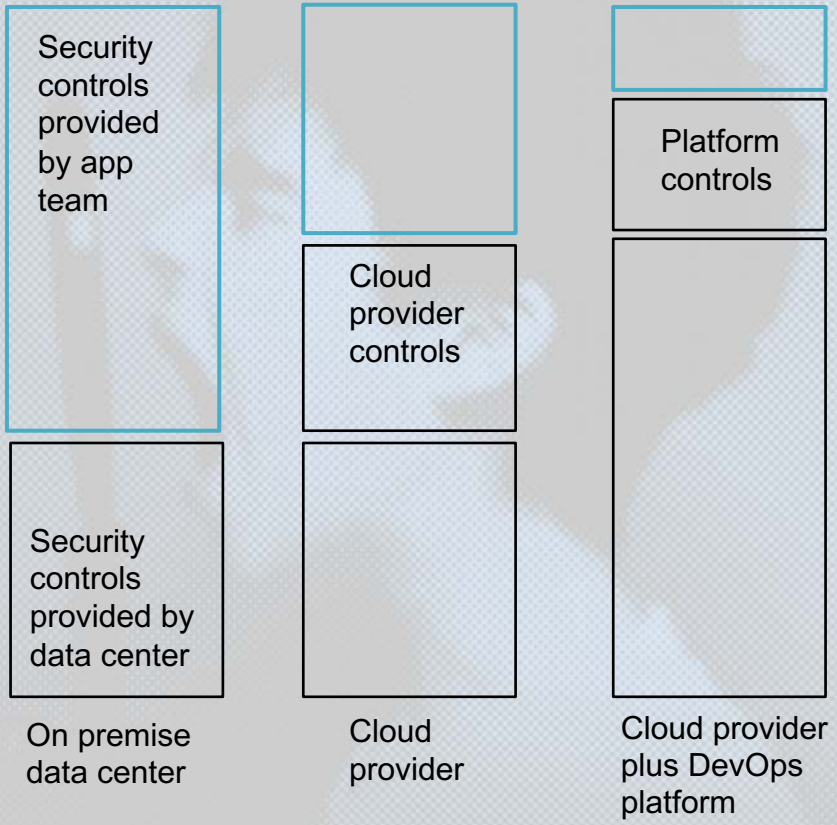
- Continued training and coaching sessions, but trying to minimize interruptions
- Education/lecture series on tools, concepts, concrete implementation steps
- Sustained executive management support and long-term vision

Challenge #4: Authority-To-Operate

- Assembling/documenting controls in the System Security Plan (SSP) is time consuming (see NIST SP 800-53)
- Sometimes takes months

Possible solutions:

- Leverage cloud solutions
- Develop DevOps-as-a-Service platform and obtain an ATO for the platform



Challenge #5: Secure DevOps tools integration

- DevOps tool chain itself also needs to be secured
- Often dozens of tools
- Introducing DevOps without security governance creates a heightened risk
- Each DevOps tool has their own privileges/credentials
- Secrets management not centralized

Possible solutions:

- Prefer solutions that offer integrated credentials management across the DevOps pipeline
- Employ a secrets management solution

Source: https://blog.devolutions.net/2018/03/why-your-business-needs-a-secure-digital-vault

Challenge #6: Secure the software supply chain

- 80-90% of software is composed of third-party libraries
- Many contain known vulnerabilities, yet are included anyway
- Many open-source libraries don’t have CVE’s filed due to burden for the developer

Possible solutions:

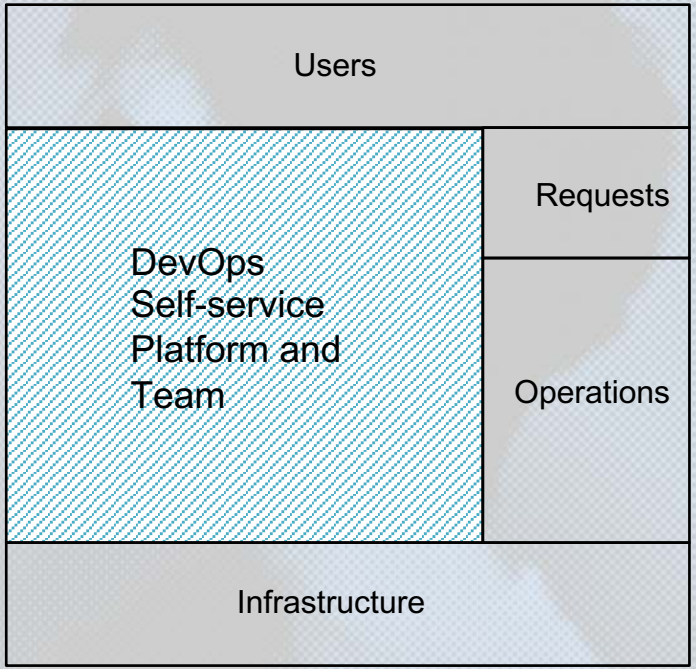
- Employ tools that scan libraries before they enter the value stream
- Employ Runtime-Application Self-Protection tools
- Leverage existing open-source intelligence when possible
- Use tools able to continuously monitor applications during development, testing, deployment and runtime

DevOps-as-a-Service team and platform

- Culmination of efforts to address above challenges
- Platform team combining all necessary skills
- Focused on building out the platform
- Created Monarch platform

Key goals and concepts

- Embedded security into CI/CD workflow
- Immutable infrastructure
- Infrastructure as code
- Self service for development teams
- Achieve platform ATO
- Ensure security compliance of deployed applications
- Consistency between environments
- Applications not tied to infrastructure



Continuously increase self service capabilities

Monarch overview

Development Teams

- Manage
- Examine
- Troubleshoot
- Monitor

go Continuous Delivery

- Source
- Build
- Store
- Deploy
- Run
- Scan

GitHub Enterprise

docker

JFrog Artifactory

Terraform

Amazon ECS

Twistlock

netsparker

Self-service security scanning

GOALS
Ensure application security compliance
Deliver more secure applications, more frequently

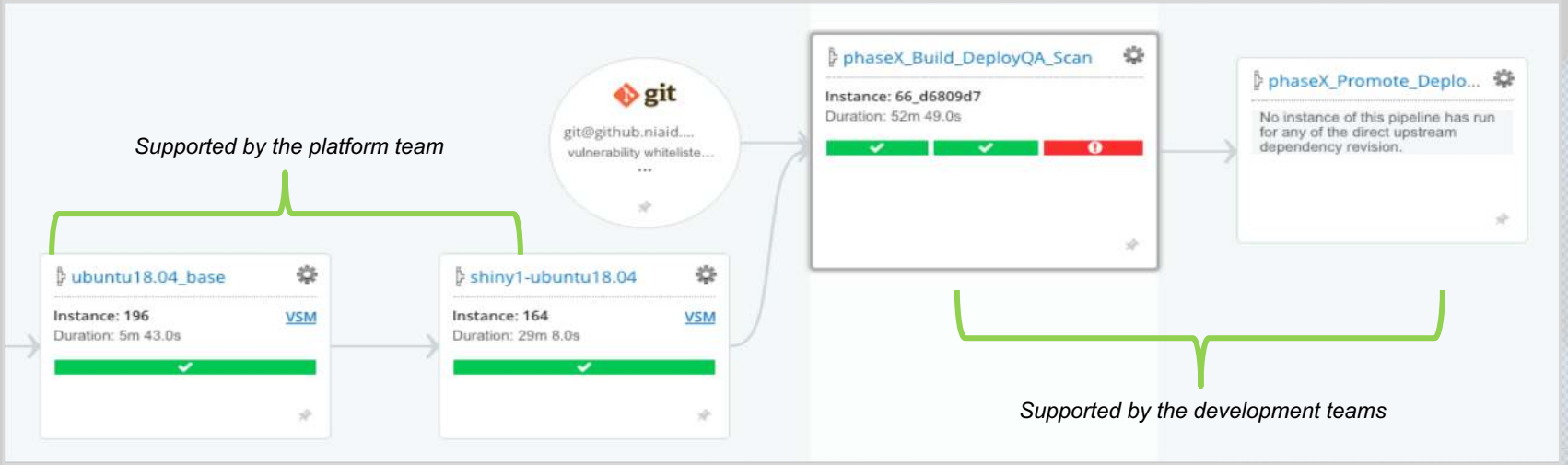
PROBLEM
Multiple hand-offs and manual gatekeeping

SOLUTION
Embed security scanning into deployment pipelines

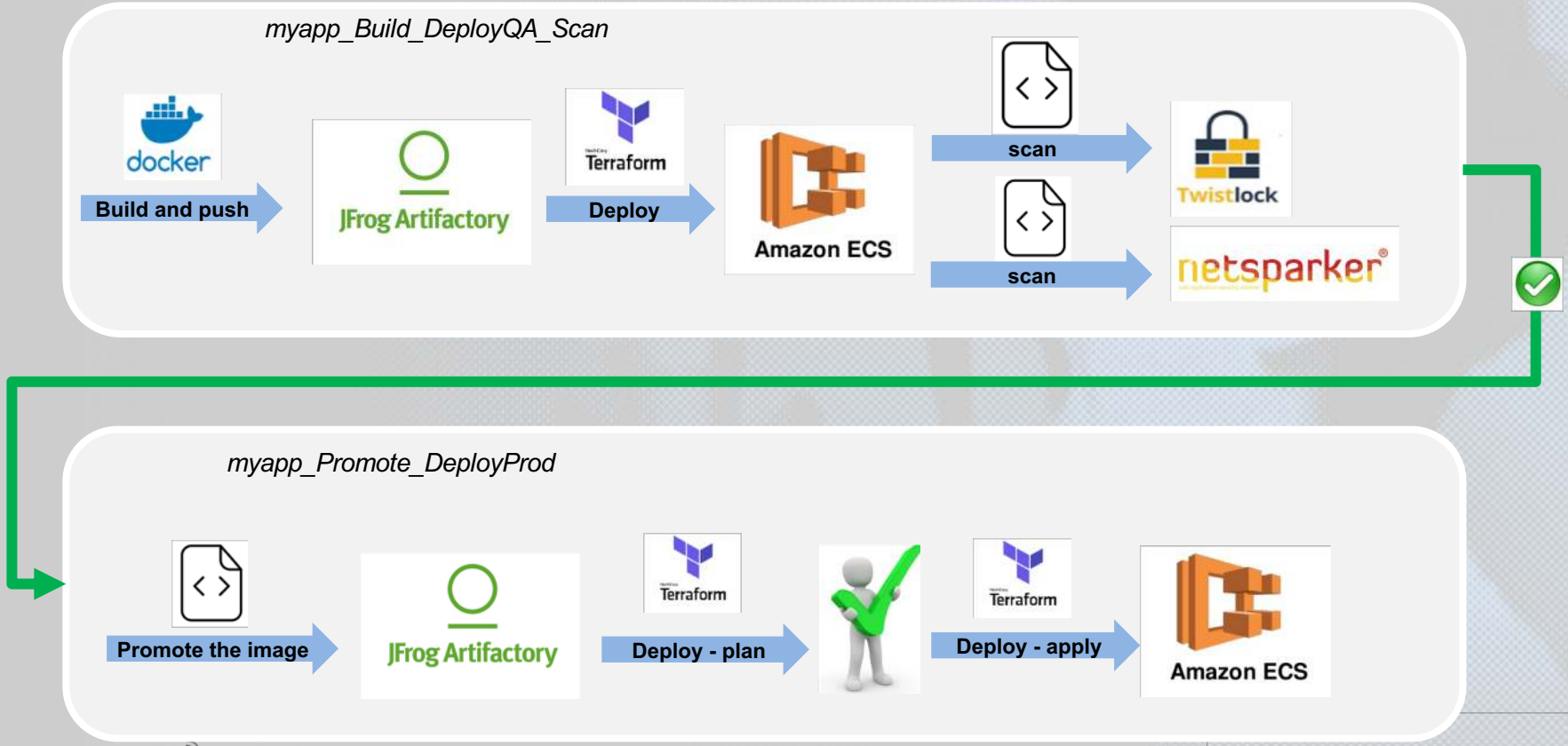
ELEMENTS
Container image scanning
Dynamic application scanning

Implementing deployment pipelines in GoCD

- Pipelines triggers - SCM repositories or another pipeline(s)
- Complex workflows via pipeline dependencies
- Streamlined pipeline creation via templates



Security scanning in pipelines



Twistlock – Docker image scanning

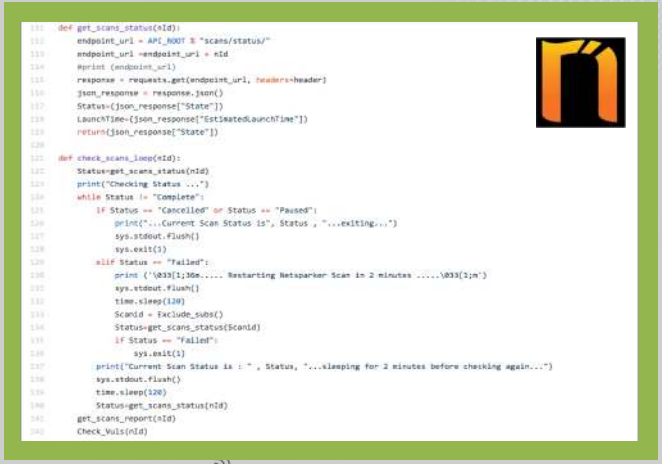
- Check software binaries, libraries, etc. for known vulnerabilities (CVEs)
- Scan application images files & packages
- Analyze Dockerfile and image metadata to detect insecure configurations:
  - running as privileged user
  - sensitive information in environment variables

```
[go] Task: twistlock_scan.rb took: 17.389s

Pulling down monarch-p                                    s:274_439ff72... got it!

Checking to see if twistlock.internal.                    is up... it is!

Grab the login token... acquired!

Starting a scan... started!

Getting your scan results in 10 seconds... got them!

Creating raw_scan_report.json and short_scan_report.json artifacts... complete!

No vulnerabilities or compliance violations detected!

Pruning old images and containers... complete!
```

```ruby
twistlock_host = ENV["TWISTLOCK_HOST"]
registry_host  = ENV["REGISTRY_HOST"]
user           = ENV["TWISTLOCK_USER"]
pass           = ENV["TWISTLOCK_PASS"]
repo           = ENV["IMAGE_REPO"]
name           = ENV["IMAGE_NAME"]
tag            = ENV["IMAGE_TAG"]
url            = "http://#{twistlock_host}:8081/api/v1"

if ( repo == "" )
  scan_repo = "#{registry_host}/#{name}"
else
  scan_repo = "#{registry_host}/#{repo}/#{name}"
end

request_body_map = {
  :host => "#{twistlock_host}",
  :imageTag => {
    :repo => "#{scan_repo}",
    :tag  => "#{tag}"
  }
}

begin
  RestClient.post("#{url}/health/images/scan",
                  request_body_map.to_json,
                  {:authorization => "Basic #{Base64.strict_encode64("#{user}:#{pass}")}",
                   :content_type  => 'application/json',
                   :accept        => 'application/json'})
```

Netsparker Cloud – web app scanning

- Detect and exploit different vulnerability types:
  - -cross-site scripting (XSS)
  - SQL injection
  - local/remote file inclusion
  - command injection
  - server-side request forgery (SSRF)
- Detect old web applications and libraries



```
def get_scans_status(nId):
    endpoint_url = API_ROOT % "scans/status/"
    endpoint_url =endpoint_url + nId
    #print (endpoint_url)
    response = requests.get(endpoint_url, headers=header)
    json_response = response.json()
    Status=(json_response["State"])
    LaunchTime=(json_response["EstimatedLaunchTime"])
    return(json_response["State"])

def check_scans_loop(nId):
    Status=get_scans_status(nId)
    print("Checking Status ...")
    while Status != "Complete":
        if Status == "Cancelled" or Status == "Paused":
            print("...Current Scan Status is", Status , "...exiting...")
            sys.stdout.flush()
            sys.exit(1)
        elif Status == "Failed":
            print ('\033[1;36m..... Restarting Netsparker Scan in 2 minutes .....\033[1;m')
            sys.stdout.flush()
            time.sleep(120)
            ScanId = Exclude_subs()
            Status=get_scans_status(ScanId)
            if Status == "Failed":
                sys.exit(1)
        print("Current Scan Status is : " , Status, "...sleeping for 2 minutes before checking again...")
        sys.stdout.flush()
        time.sleep(120)
        Status=get_scans_status(nId)
    get_scans_report(nId)
    Check_Vuls(nId)
```

```
[go] Task: Netsparker-Scan took: 13m 2.471s
my yaml_file is  monarch.yaml


Default Scan policy : Platform.2018.05

New Scan ID is  8dc6a70b-50d4-49fb-1def-a9750320125d
Scan requested for  https://m[redacted]/
..... Starting Netsparker Scan .....
Checking Status ...
Current Scan Status is :  Queued ...sleeping for 2 minutes before checking again...
Current Scan Status is :  Scanning ...sleeping for 2 minutes before checking again...
Current Scan Status is :  Scanning ...sleeping for 2 minutes before checking again...
Current Scan Status is :  Scanning ...sleeping for 2 minutes before checking again...
Current Scan Status is :  Scanning ...sleeping for 2 minutes before checking again...
Current Scan Status is :  Archiving ...sleeping for 2 minutes before checking again...


**************  Calculating Vulnerabilities  ********************

Original Count of vulnerabilities =  0

************ This website has no whitelisted items ***************


***** Recalculating ....(Any Whitelisted Items will be deducted) *****

New Total Vulnerabilities  =  0
```

Scanning configuration

monarch.yaml:

- Stored in the root of an application repo
- Contains application configuration for pipelines stages
- May include white-list items for web application and Docker image scans

```
name: myapp

docker_build:
  variables:
    IMAGE_REPO: myrepo
    REGISTRY_USER: mylogin

webapp_scan:
  variables:
    NS_URL: 'https://myapp.qa.domain.net/'
  whitelist:
    - Missing Content-Type Header:
    - "https://myapp.qa.domain.net/__sockjs__/n=AETttrbrbCun47684/054"

docker_scan:
  cve_whitelist:
    - CVE-2018-1234  # False positive
    - CVE-2018-5678  # Not applicable
```

Results and impact

CURRENT STATE

- Early vulnerability detection
- Quick remediation response (mitigating Spectre and Meltdown for all apps took less than 5 days)
- Frequent updates in production are encouraged
- Multiple parties involved in security auditing

IN THE WORKS

- Code quality monitoring

sonarqube

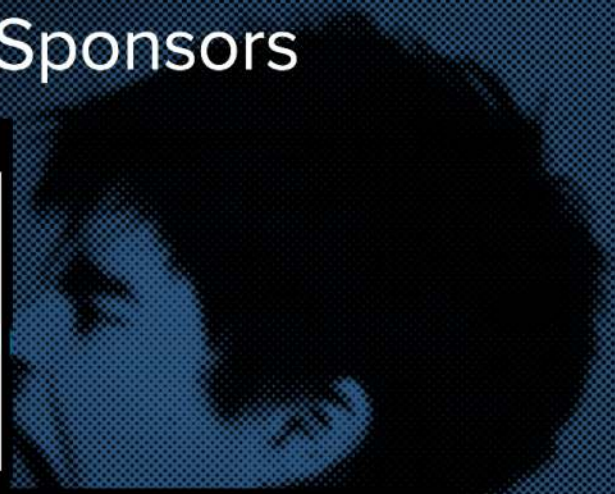
Conclusion

- Facing large challenges in securing software production and delivery
- We outlined six of them and how our team together with our customer addressed them
- It’s a continuous improvement process and never ends
- Application security governance program for coordination highly recommended

Source: https://commons.wikimedia.org/wiki/File:The_road_ahead_(2046262670).jpg

- Thank you for joining us!

Thank You All Day DevOps Sponsors



Platinum Sponsors

Sonatype

cloudbees

Gold Sponsors

puppet

GitLab

XebiaLabs
Enterprise DevOps

GENERAL DYNAMICS
Information Technology

cisco DEVNET
developer.cisco.com

Carnegie Mellon University
Software Engineering Institute

CHEF

SCALED AGILE

Media Sponsors

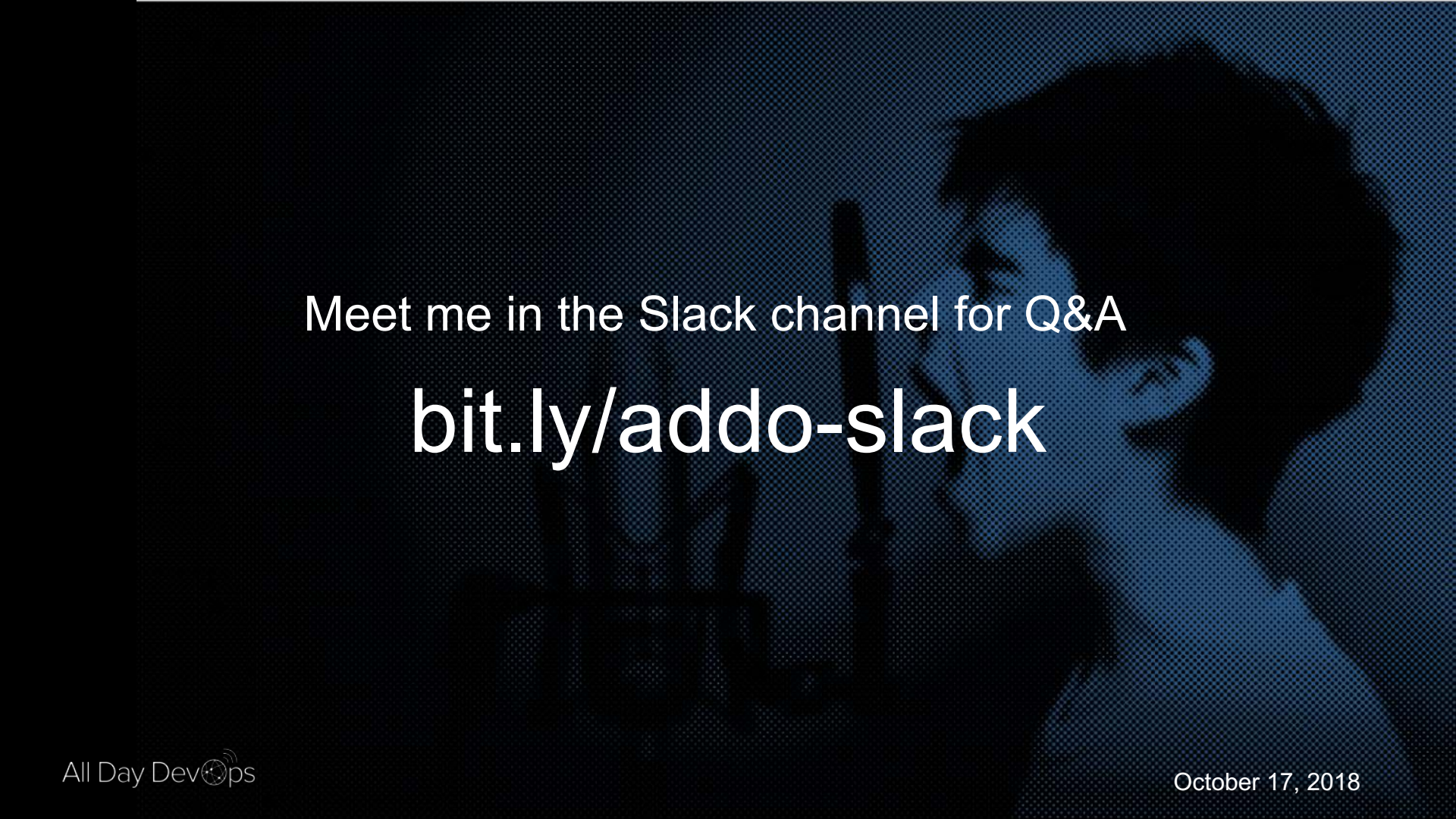
TechBeacon

DevOps.com

DZone

Solutions Review

Thank You All Day DevOps Supporters


Meet me in the Slack channel for Q&A
bit.ly/addo-slack
All Day DevOps
October 17, 2018