RSAConference2022
San Francisco & Digital | June 6 – 9

TRANSFORM

SESSION ID: **DSO-RO2**

What Executives Need to Know about CI/CD Pipelines and Supply Chain Security

Dan Cornell

Vice President, Product Strategy
Coalfire
Twitter: @danielcornell

Disclaimer

Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the presenters individually and, unless expressly stated to the contrary, are not the opinion or position of RSA Conference LLC or any other co-sponsors. RSA Conference does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented.

Attendees should note that sessions may be audio- or video-recorded and may be published in various media, including print, audio and video formats without further notice. The presentation template and any media capture are subject to copyright protection.

Agenda

- What is a CI/CD pipeline and why should I care?
- Follow a code change to understand exposure
- Put what we've learned today into practice

What is a CI/CD Pipeline and Why Should I Care?

Starting Question

Who here has a software development background?

(Modern stuff – COBOL and FORTRAN don’t count)

What is a CI/CD Pipeline?

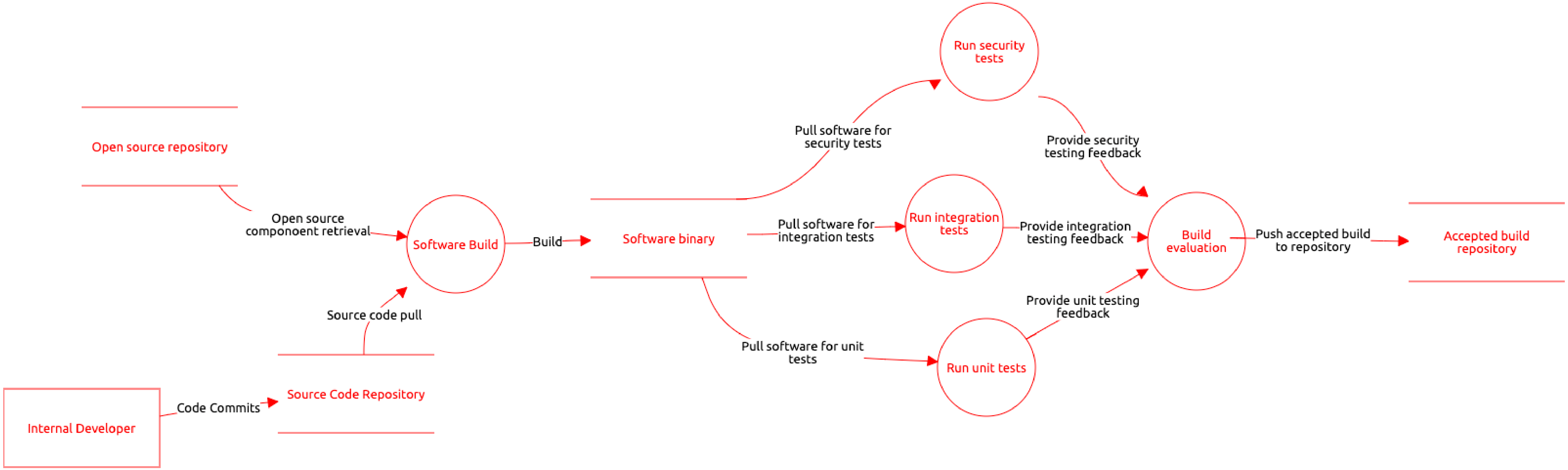
- Continuous Integration /Continuous Delivery
- Set of steps ("pipeline") to deliver a new build of software
- Focus on *automation* and *repeatability*

Answers The Question

Is this software build acceptable to deliver?

This practice and set of associated tools is a key component
in any team practicing a DevOps/DevSecOps culture

Example CI/CD Pipeline Dataflow


Open source repository
Open source componoent retrieval
Software Build
Build
Software binary
Pull software for security tests
Run security tests
Provide security testing feedback
Pull software for integration tests
Run integration tests
Provide integration testing feedback
Build evaluation
Push accepted build to repository
Accepted build repository
Pull software for unit tests
Run unit tests
Provide unit testing feedback
Source code pull
Source Code Repository
Code Commits
Internal Developer

Why Should I Care?

- This is where your software gets built and delivered

- Lots of sensitive data involved in the process
- Lots of avenues in for malicious parties
- Responsible for your output
  - Impacts software running in your environment
  - Impacts software you deliver to customers

- If this gets compromised, lots of other things get compromised
  - Tremendous attacker leverage

Impact of Compromise

Confidentiality

IP disclosures

Leaked secrets

Vulnerability disclosures

Integrity

Backdoors

Other unwanted behaviors

Compromised cryptography

Availability

Release delays

Inability to push fixes to production

Notable Incidents Related to CI/CD Pipelines

- **Solar Winds**
  - Compromised server in build infrastructure
  - Led to SolarWinds publishing modified/malicious software builds
- **CodeCov**
  - Compromised container server
  - Led to compromise of information stored in CodeCov's customers' build environments
- **Helped to spawn US Executive Order, other government activity**
  - (and even more annoying vendor onboarding questionnaires)

The Fundamental Disconnect

We have developed tremendous tooling and automation that allows us to create secure, reliable software at a scale not previously considered.

The way we deploy and deliver this tooling puts us in a situation where we can’t actually trust anything that comes out of it – Oops.

RSAConference2022

Follow a Code Change to Understand Exposure

Follow a Code Change

Change code

- Check out code (and make changes)
- Create merge request
- Review merge request

Run build

- Pull code
- Pull open-source components

Automated testing

- Unit tests
- Acceptance tests
- Security tests
- Other tests

Accept build

Push to distribution

Distribute software

Follow a Code Change (continued)

Questions to ask at each stage

- Where is the server/service that performs this step?
- Where did the data needed for this step come from and how was it protected in transit?
- How did the server/service authenticate itself to the data provider?
- How does this user/system initiate the next step?

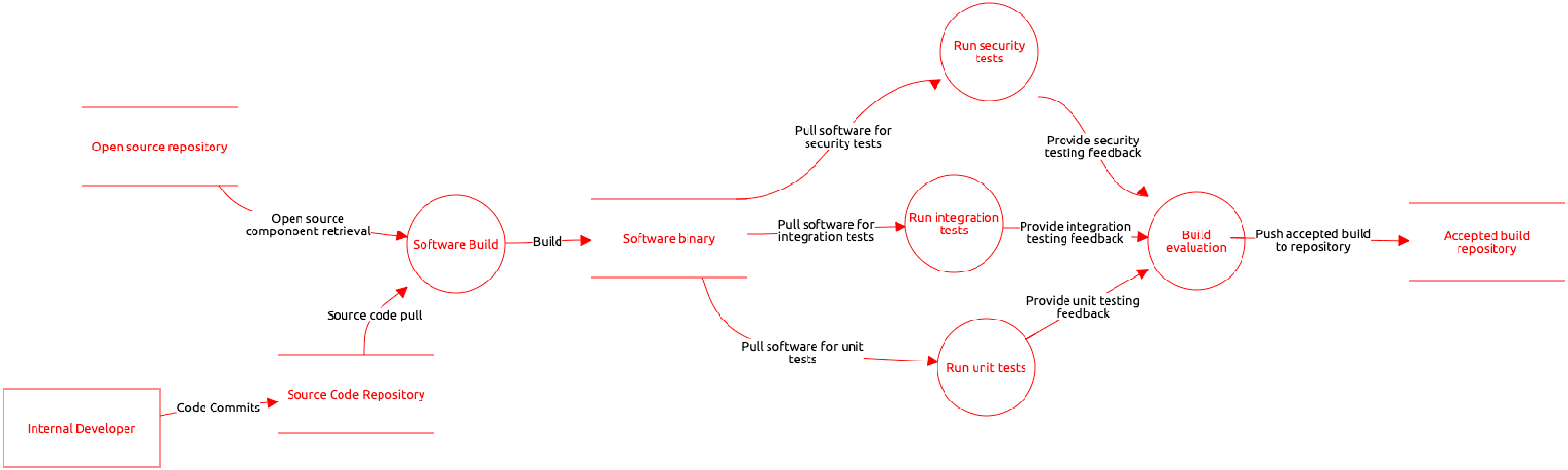
Have them show. Don't let them just tell.

- Look at what the UI developers look at
- Don't stop asking questions until you get an (actual) answer
- The DevOps rep will learn some valuable things from this exercise as well

Fractal Attack Surface

The closer you look, the more you'll find

Example CI/CD Pipeline Dataflow



Open source repository

Open source componoent retrieval

Internal Developer

Code Commits

Source Code Repository

Source code pull

Software Build

Build

Software binary

Pull software for security tests

Run security tests

Provide security testing feedback

Pull software for integration tests

Run integration tests

Provide integration testing feedback

Pull software for unit tests

Run unit tests

Provide unit testing feedback

Build evaluation

Push accepted build to repository

Accepted build repository

What an Executive Needs to Know

- **What pipelines does my organization have?**
  - Similar question to “What applications does my organization have?”
- **What data is in play and where does it go?**
  - Source code, test data
- **What controls do I have in place?**
  - Data protection, authentication, authorization

General or Overarching Concerns

- **Lots of network traffic**
  - Protect with TLS to guard Confidentiality, Integrity
- **Lots of authentication points/IAM concerns**
  - Often multiple methods/avenues per system: interactive + API (+ other?)
  - User <-> System
  - System <-> System
  - IS there a comprehensive IAM paradigm in place, or a combination of overlapping approaches?

Identity and Access Management

How does the IAM solution manage and rotate access tokens to mitigate against cracking?

Does the solution allow a forced change of signature algorithm?

What information is included in an authentication claim?

How does the organization ensure the endpoint services verify all the identity data it should?

General or Overarching Concerns

- **Data storage concerns**
  - How is data at rest protected? (Especially file/block storage)
- **What are your unknown-unknowns?**
  - External services and providers that are (generally) unknown but part of the process
  - Any developer with a GitHub account is now their own purchasing agent

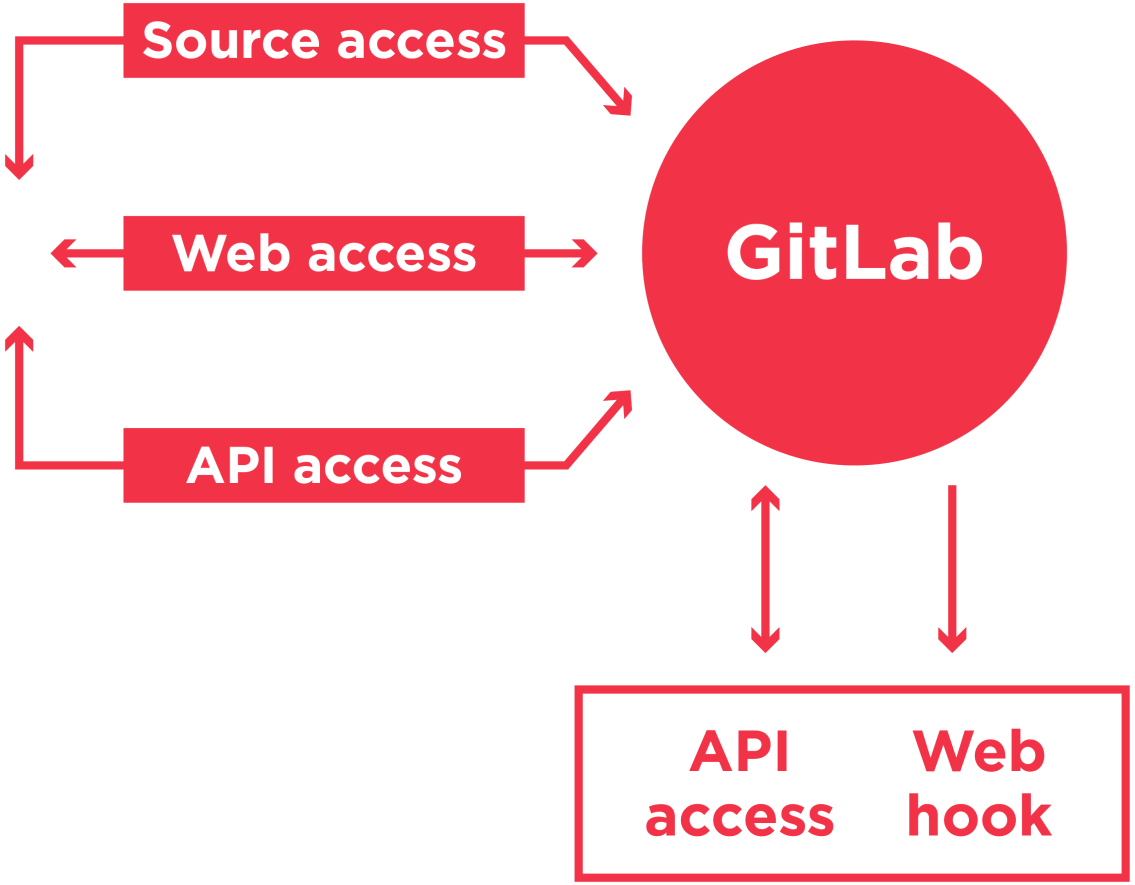
Storage

Does the organization maintain a clear data classification policy?

What cryptographic solutions are in-place to adhere to this policy?

What process verifies adherence?

Source Repository and Workflow Engine



Source Repository and Workflow Engine

- **Source of (custom) code**
  - In-house developers
  - 3rd party development teams

- **Likely different risk profiles**
  - Do contributions go through different workflows?

Source Repository and Workflow Engine

Do the repository and workflow engine provide the interfaces for source/build extraction and ticketing necessary a touchless SAST solution?

Are known vulnerabilities integrated into the workflow in a manner that allows outside stakeholders to monitor and verify resolution?

Source Repository and Workflow Engine

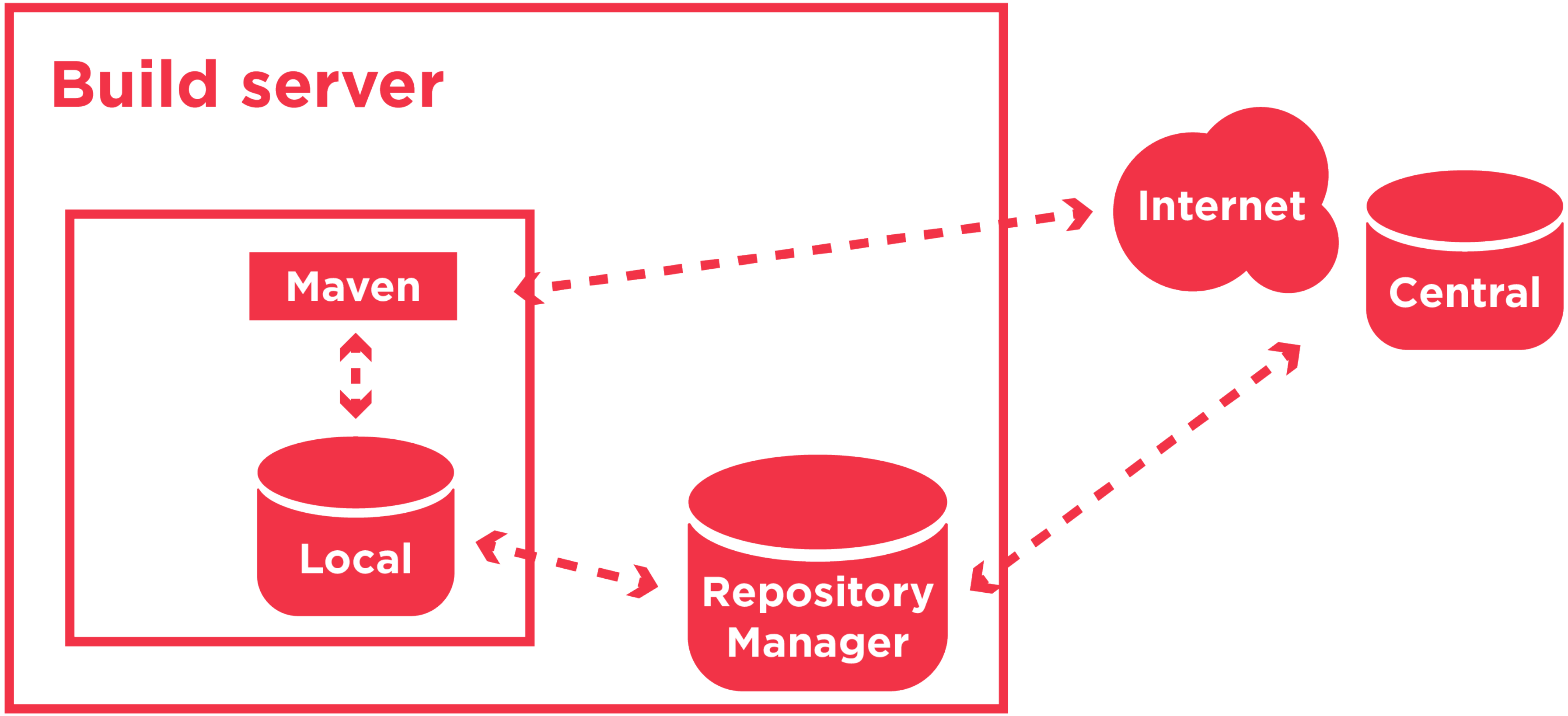
- **Authentication**
  - Developers authenticating to push/pull code
  - Developers authenticating to evaluate/approve merge requests
  - External system API access
  - Webhooks

- **Additionally for hosted repositories/workflows**
  - GitHub: Apps and Actions
  - GitLab: Integrations
  - Both: External collaborators

Executive Action: Source Code

- Know where your source code is being stored and managed
- Know how you protect access to source code repositories

Open-Source Component Management



Open-Source Component Management

- **Are you:**
  - Directly pulling from central repositories?
  - Proxying requests through a remote repository?

- **Ability to enforce policies**
  - Known security vulnerabilities in open-source components
  - License restrictions on open-source components

Dependency Management

Does the dependency management solution control the application dependencies propagated to production, or does the production team defer to artifacts provided by developers?

Does the solution adequately encompass all imported application assemblies and recognize UI frameworks?

Does it similarly account for open-source projects built alongside organization’s developed source?

Open-Source Backdoor Concerns

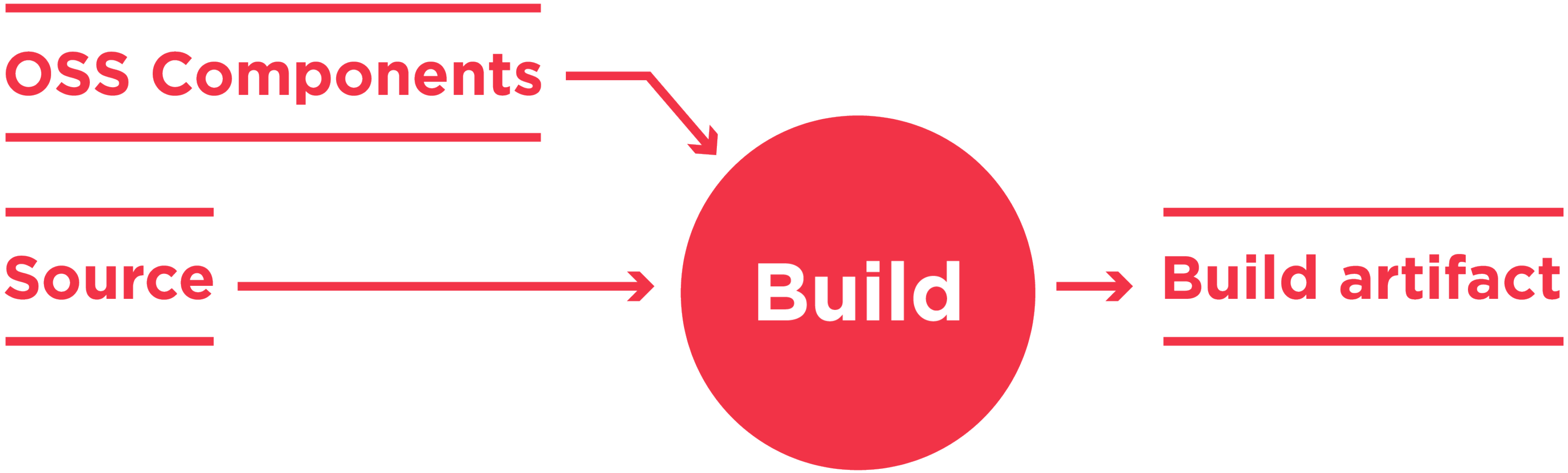
- **PHP example**
  - https://arstechnica.com/gadgets/2021/03/hackers-backdoor-php-source-code-after-breaching-internal-git-server/
- **Linux kernel example**
  - https://www.techrepublic.com/article/linux-kernel-security-uproar-what-some-people-missed/

- **Not realistic to detect at scale**
  - Package maintainer don’t even have the bandwidth
- **Must be able to *respond***
  - Pulling from remote repository vs. directly from central
  - Policies on acceptable – and unacceptable - versions

Executive Action: Open-Source

- Know what open-source software is in use
  - Software Bill of Materials (SBOM)
- Understand what controls you have on how new open source is added

Build Management



OSS Components

Source

Build

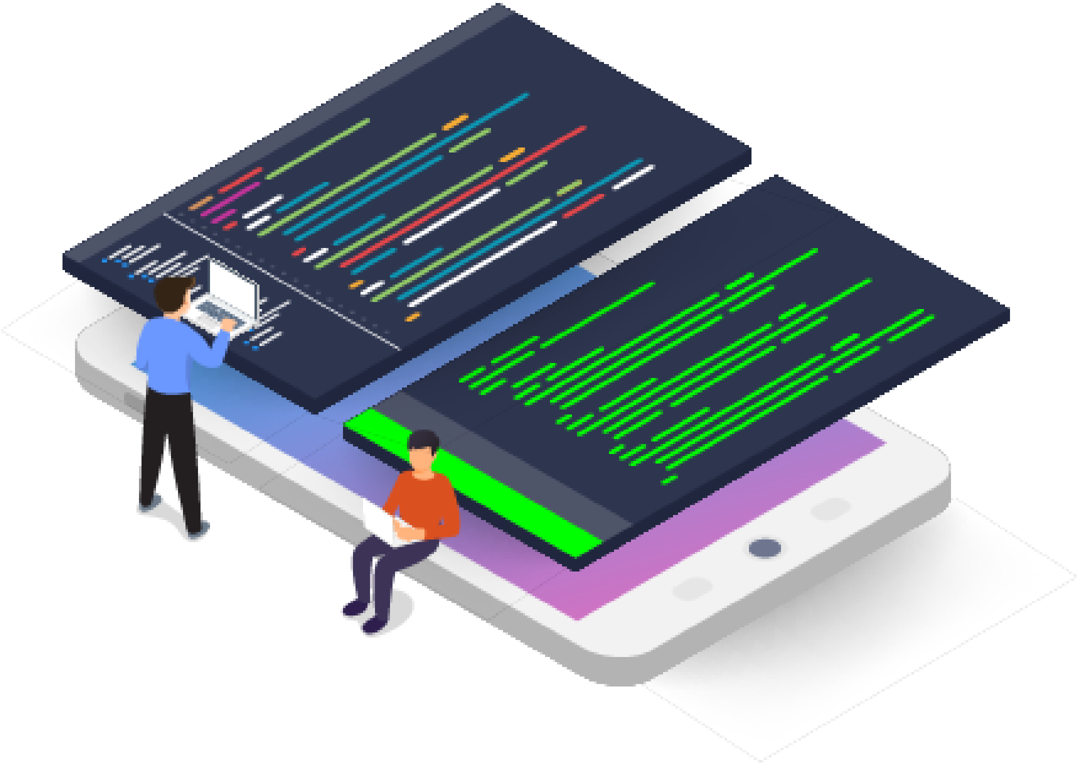
Build artifact

Build Management

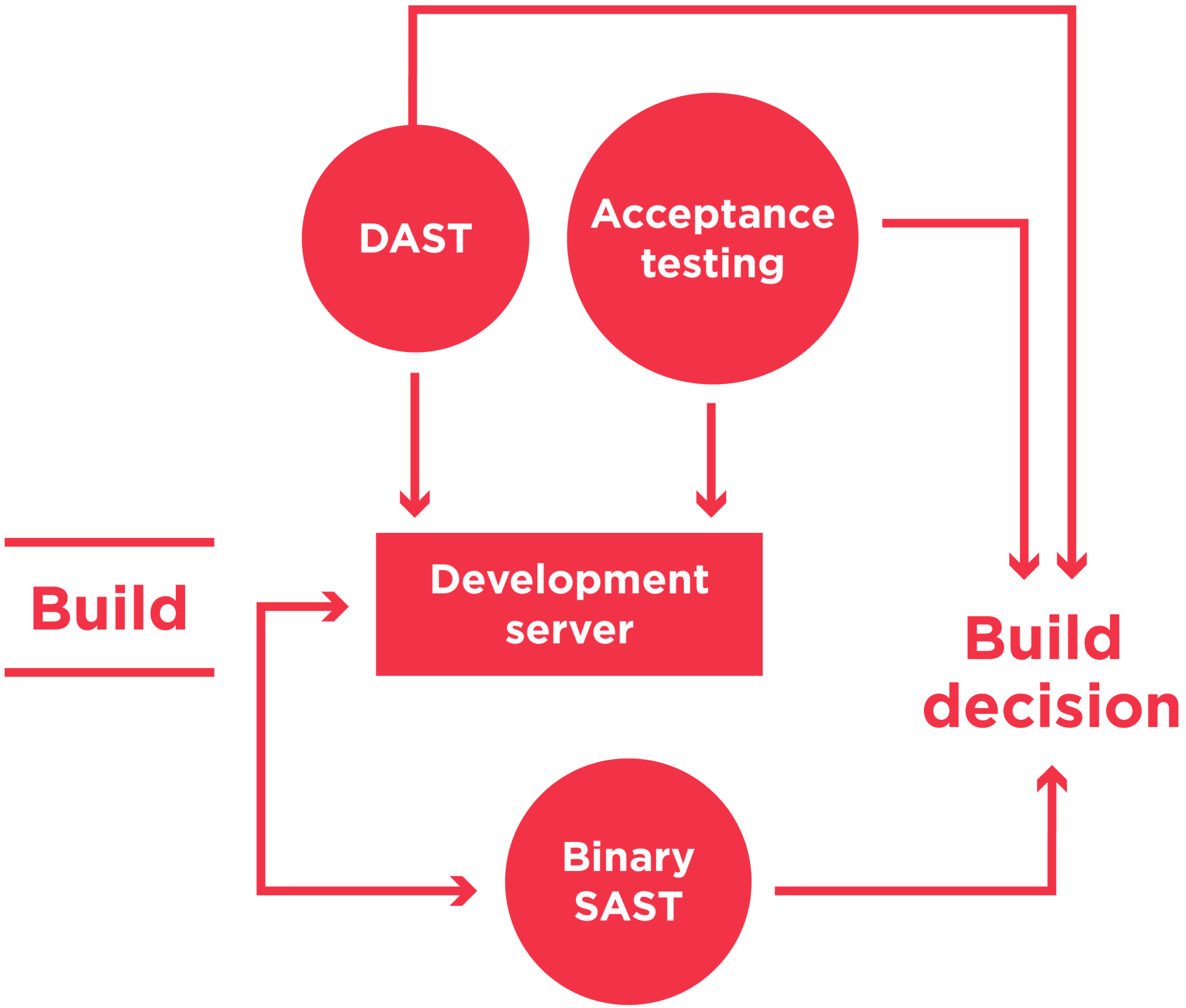
- Combining source code and open-source components to create a new build
- How often are builds run?
  - Every merge request?
  - Accepted merge requests?
  - On changes to specific branches/tags?
- Maven plugins – yet another source of code you can't trust that runs in your environment!

Executive Action: Build Management

- How often are builds being run?
- What build plug-ins are in use?
  - Where did they come from?
  - What data do they access and where do they send it?



Automated Testing



Automated Testing

- **Most common types of testing:**
  - Unit testing
  - Acceptance testing
- **Other common analysis:**
  - Code quality (“smells”)
  - Code metrics (complexity, etc.)
  - Code coverage

Automated Testing: Security Testing

- **SAST**
  - What is being analyzed: source or binary?
    - For a given language, is "binary" even a thing?
  - Where is the analysis being performed?
  - Where are the results being stored?
- **DAST**
  - Where is the server being tested?
  - Where is the test traffic being generated? Is it being proxied?
  - Where are the results being stored?

Static and Dynamic Scanning Tools

Are the current or planned DAST and SAST tools adequate for the technologies they will need to cover long-term?

Does the process for responding to vulnerability alerts prompt root cause analysis and tool customization when appropriate?

Automated Testing: Security Testing

- **IAST**
  - Where is the server being tested?
  - Where is the test traffic being generated? Is it being proxied?
  - Where are the results being stored?
- **SCA**
  - Where is the analysis being performed?
  - Where are the results being stored?

Risk and Vulnerability Management

Is the vulnerability management solution configured to allow all appropriate parties access needed to fulfill their responsibilities without allowing any lapses in integrity, such as a developer suppressing vulnerability findings without the security team's knowledge?

Security Testing and Backdoors

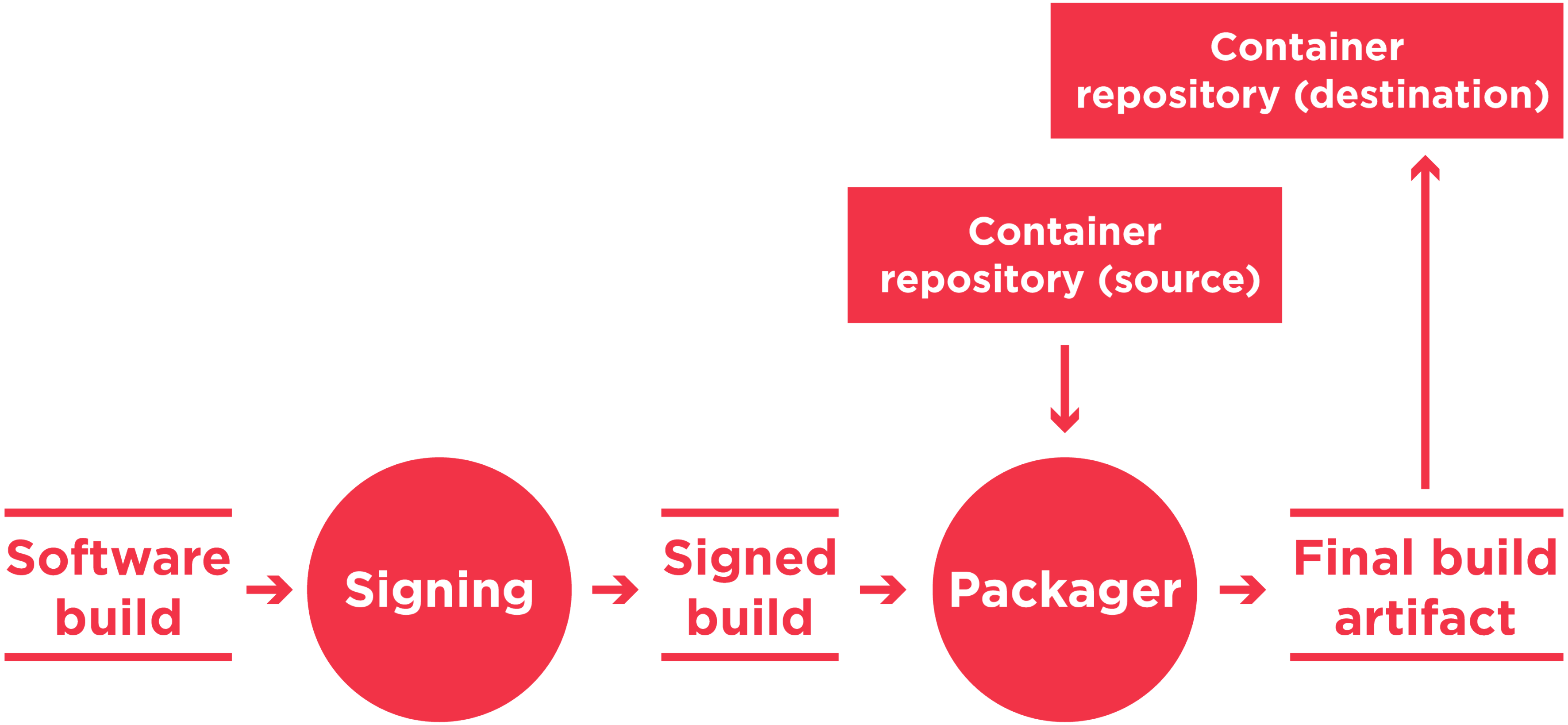
- **Best backdoor? Normal-looking security vulnerabilities**
  - “Oops, sorry I introduced that [SQL injection, missing authorization check, etc.]”

- **Other types of backdoor detection**
  - Static analysis – see the theoretical behavior of the application
  - Look for suspicious behavior patterns
  - References:
    - https://www.acsac.org/2007/workshop/Wysopal.pdf
    - https://www.veracode.com/sites/default/files/Resources/Whitepapers/static-detection-of-backdoors-1.0.pdf
    - https://owasp.org/www-pdf-archive/Protecting_the_Enterprise_-_Software_Backdoors.pdf
    - https://owasp.org/www-pdf-archive/Protecting_Your_Applications_From_Backdoors.pdf

Executive Action: Testing

- What sort of test coverage do you have?
- What 3rd parties are involved in the process?
- Where is source code being sent?

Software Packaging and Distribution



Container repository (destination)

Container repository (source)

Software build → Signing → Signed build → Packager → Final build artifact

Software Packaging

- **Monolithic application vs. microservices applications**
  - Application binaries vs. application binaries combined with containers
- **Source container management is very similar to open-source component management**
- **Code signing is important**
  - But signed malicious/vulnerable code is still malicious/vulnerable code

Container Registry

Are application containers configured or approved by security architects?

Are these architects aware of the scope of risk the configuration should account for, from the operating system to application runtime engines to web containers?

Do the processes around the creation and propagation of containers control any attempts to deviate from approved configurations?

Software Distribution

- **Now the software should be ready to distribute to customers**
  - Internal or external
- **How is the software distributed?**
  - Binary
  - Binary + container
  - Binaries + containers + orchestration
- **How are builds verified?**
  - Checksums?

Executive Action: Packaging and Distribution

- What container images are in use?
  - And how are you managing vulnerabilities
- How do you sign/protect software builds?
- How do you control who can access builds?

RSAConference2022

Put What We’ve Learned Today Into Practice

Applying This Information

- **Next week you should:**
  - Start enumerating the CI/CD pipelines within your organization
- **In the first three months following this presentation you should:**
  - Have a completed list of pipelines
  - Do walkthroughs of three to five pipelines to identify patterns
- **Within six months you should:**
  - Develop and provide guidance to Dev(Sec)Ops teams about pipeline security standards and practices

Questions

Dan Cornell

Email: daniel.cornell@coalfire.com

Twitter: @danielcornell

LinkedIn: linkedin.com/in/dancornell/