



Moving the AppSec program from hurdle to sprint

**Balachandra Shanabhag “Bala”
March 2, 2023**




Logistics

- Slides available at <https://github.com/matureappsec/snowfroc23> and also will be available on the snowFROC webpage
- All questions, comments and feedbacks are welcome
- Popular and not so popular opinions in entire presentation are my own



Whoami

- Doing random Cybersecurity stuff in enterprise space for 15 years now
 - Security generalist in a startup environment aka Staff Security Engineer (Cohesity)
 - Built secure security and networking gears and even physically break them in pentest (Juniper Networks)
 - 2 AppSec program bootstraps
- Social connections : 



Next 55 minutes

- Security is accepted as business enabler
- If security controls are hurdle to the business, we are the risk to the business
- Lets minimize overall AppSec risk by allowing some acceptable risks and mitigating the rest



Agenda

- Mature AppSec Program Walkthrough
- Top 5 Developer AppSec Hurdles
- AppSec Practices from Hurdles to Sprint
- Q n A

Mature AppSec Program



Plan and Design		Build and Test		Deploy
AppSec Policy				
Security Requirements	SAST	DAST/IAST	Change Management	
	Secret Scanners	Container Scans	Runtime Security	
Threat Model / Abuse Cases	SCA	Infra Scans	CSPM	
	IAC Security	Manual/ Automated security tests	Security Operations	
	Crypto Review	CD/CI security	Bug Bounty Programs	
Compliance Requirements	Secure Code Reviews	Fuzzing	PenTesting	
		Artifacts Scans		
Security Champions Program				
Application Security Trainings				



Relatable ?

Source :

<https://www.shutterstock.com/image-photo/los-angeles-april-25-2019-dmv-1433374145>



AppSec practices have multiplied at least 3X, but only few new members added to AppSec team.



Paved Roads

Term coined by Netflix for adoption of default security controls by developers

- Requires central team to build security tools and processes
- Enables developers to ship things quickly and efficiently, while maintaining an appropriate level of security (Secure by Default)



DevSecOps

Shifting security left with CD/CI pipelines*

- Repeatable, measurable and efficient
- Helps developers “do the security things” through automated pipelines
- Supports most AppSec controls in build, test and deploy SDLC phase

* Definition trimmed down to the scope of this talk



Top 5 Developer AppSec Hurdles

FALSE POSITIVES



No prize for guessing this one !!

FALSE POSITIVES EVERYWHERE

makeameme.org



False Positives

Even in mature well tuned AppSec practices false positives are common

- Define Scope of each AppSec practice based on applicable threat actors and acceptable risk
- Enforce hygiene categorize code, data, infra
 - Test code, internal / research, Dead code etc
 - Assets hosting customer data, Internal System etc
- Developer self service exception workflow for each control based on defined acceptable risk
 - Example : non critical out of context SAST finding, SCA wrong matches, Low confidence DAST findings



False Positives

Trust But Verify

- All exception workflows should be auditable by AppSec team
- Critical risks should not be part of exception workflow
 - Example : Working cloud credentials in secret scanners, Internet exposing IAC findings etc
- False negative rate < acceptable risk



Yet another library to
patch ?



Security Fatigue

Too many asks from security team

- False positive reduction will help with reducing the noise
- Prioritize issues above acceptable risk over fix-all
- Vulnerability intelligence feed over CVSS base scores

Risk of CVSS 9.8 may not always be greater than CVSS 5.1

- De dupe the security issues from different tools
- Start small and scale right



I Have Enough Tools

Said No Security Engineer Ever

Source :

<https://www.shutterstock.com/image-photo/laptop-on-work-table-diy-tion-271173740>



Tooling Overload

Don't bring developer to security tools, take security to developers

- Integrate the security tools to developer workflow
- Evaluate the tools per tech stack, best in market may not be best for you
 - Use security communities for quicker bake off and find right fit (hint: OWASP global slack group)
- Monitor tool effectiveness continuously
- Keep the tooling to essentials

In some organization lack of tooling is an hurdle for AppSec effectiveness.



Lack of developer Security Expertise

Tell me and I forget ... involve me and I learn

- Document and publish user guide and FAQs for each AppSec practices
 - Follow agile, new project is not complete unless in-depth documentation is published
- Publicise the applicable threat actors
- Internal AppSec top 10 similar to OWASP top 10
- Fund and encourage the internal bug bounty



Security and Development Synergy

I like new features.. We like secure new features..

- Define MVSP (Minimum Viable Secure Product)
 - Minimum security baseline each new applications have to meet (ref: <https://mvsp.dev/>)
- Communicate in threats and risks instead of security jargons
 - Library used has SQL injection CVE-2024-22222 ❌
 - If unpatched there is a high risk of data exposure ✅
- Developer lead threat model.



Caution!

Some non traditional security risk parameters to consider

- Reputation risk
- Compliance risk
- Hard Contracts
- Continuous updation of risk profiles based on evolving threats



Top 5 Recap

- False Positives
- Security Fatigue
- Tooling Overload
- Lack of Developer security knowledge
- Security Dev synergy



AppSec practices Hurdles to Sprint



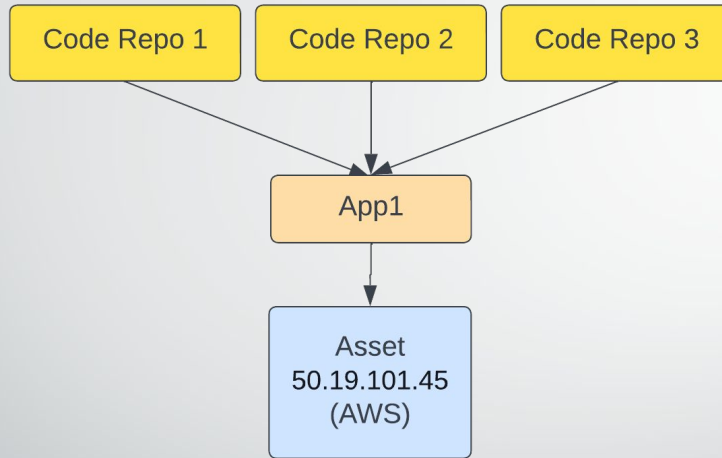
Acceptable risk examples *

- For all code repos unverified cloud credentials can be ignored.
- Other than 'cloud credentials' managed secrets are acceptable in test code.
- Use exploitability metrics for all SCA/Container/Infra scan finding risk (re-)classification.
- Any unauthenticated OSS vulnerability identified for authenticated flows can be re-classified for lower risk.
- All low risk vulnerabilities (re-classification) are acceptable.
- Non context infra secrets can be ignored. Example : mongodb password discovered in a non mondo codebase / app.

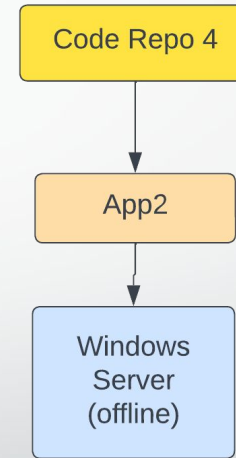
*Only examples, need customization based on organization/app
cyber risk appetite



Building the Inventory



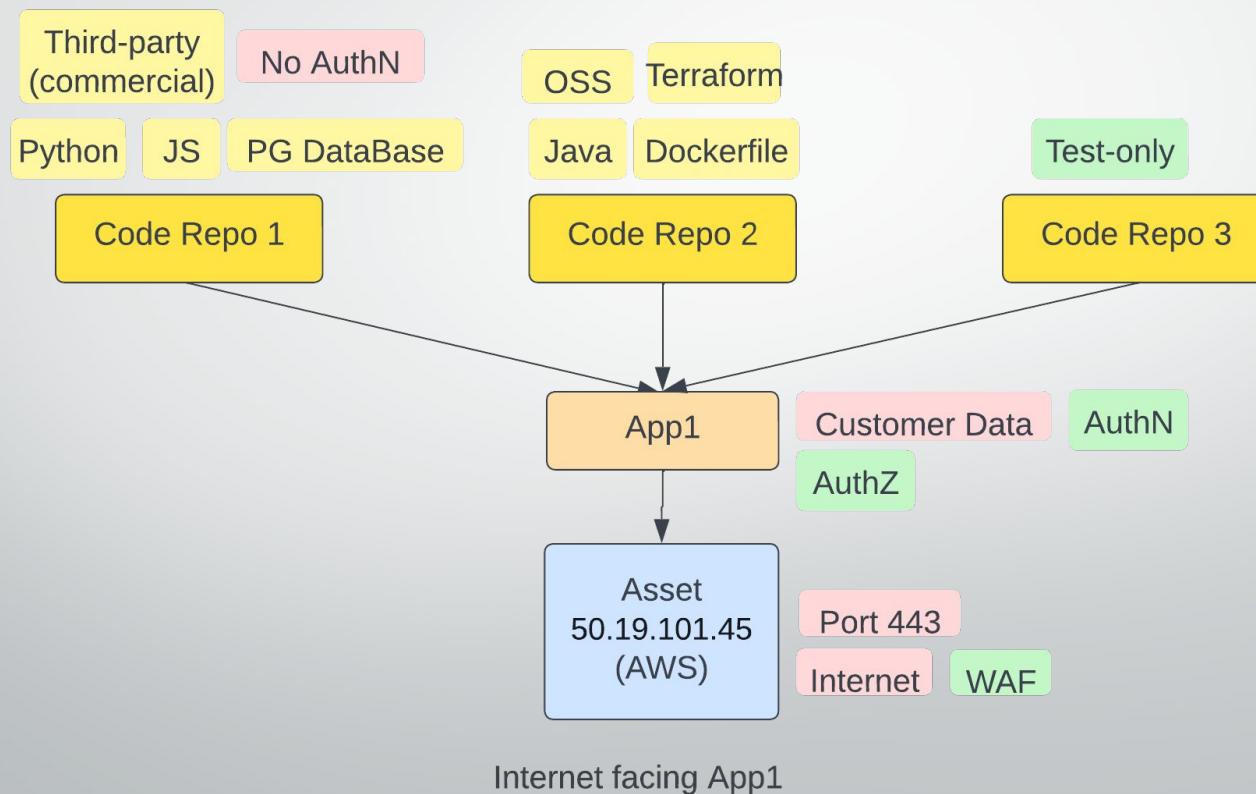
Internet facing App1



Offline windows only App2

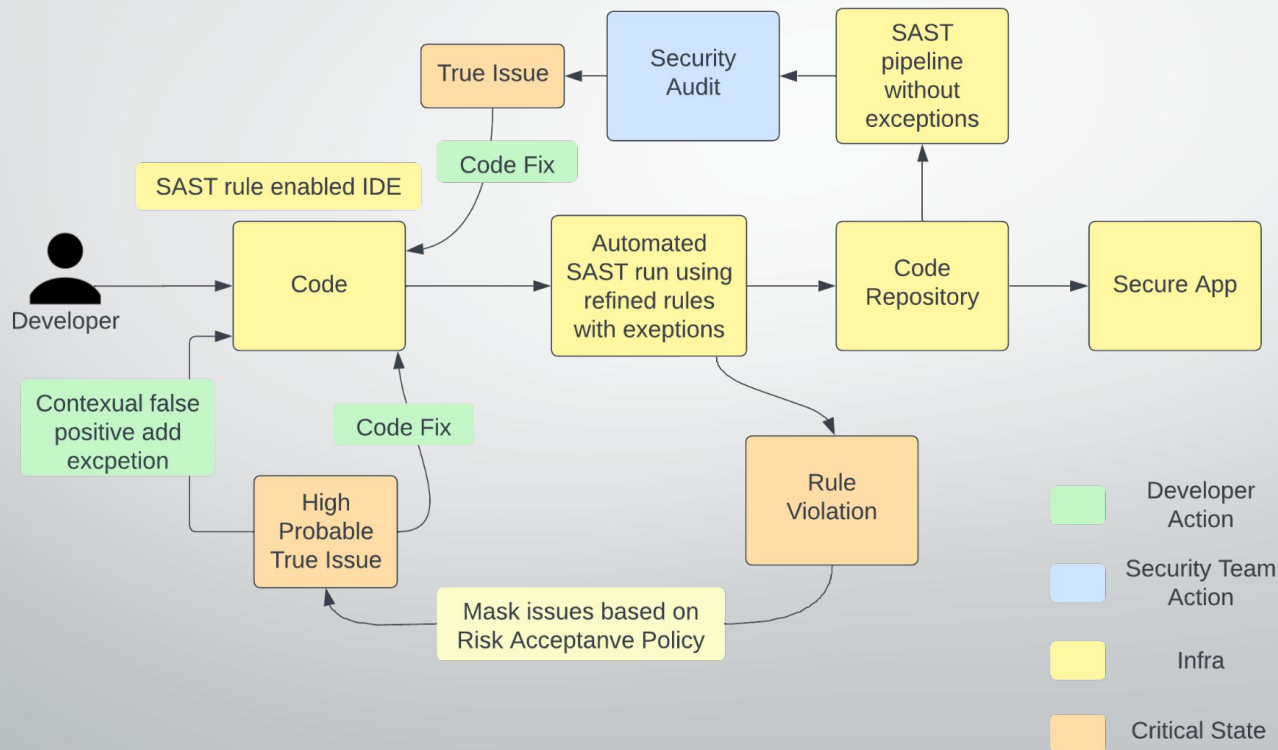


Risk label the repos, apps and assets





SAST pipeline

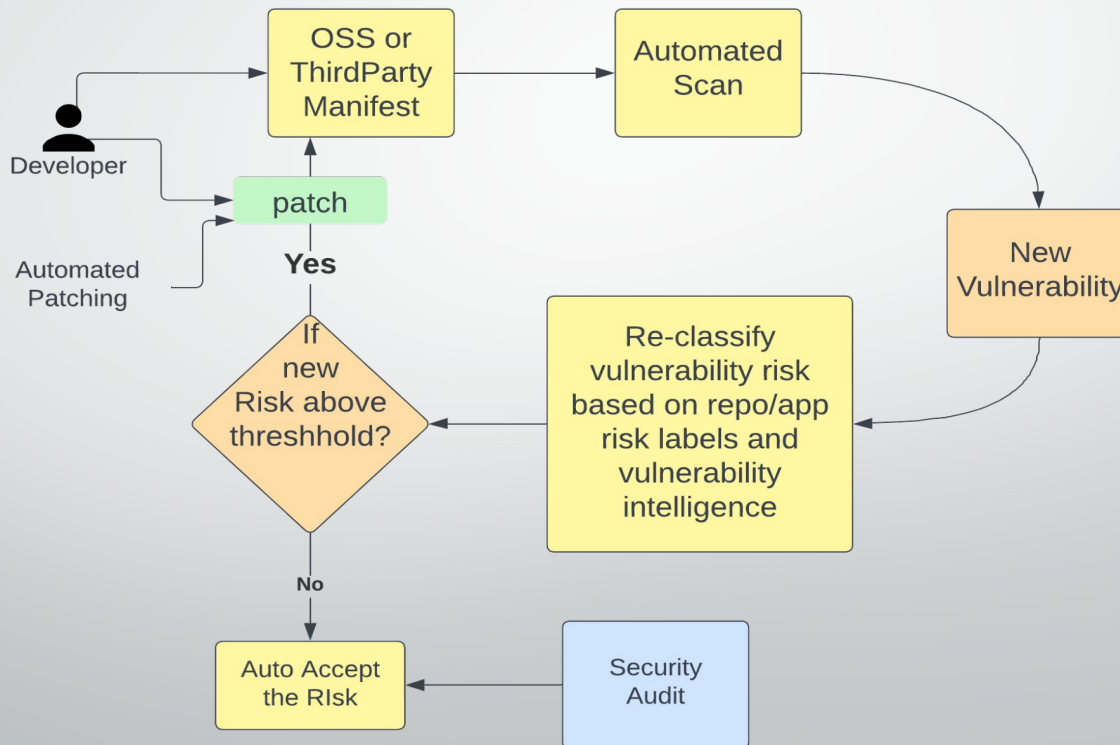




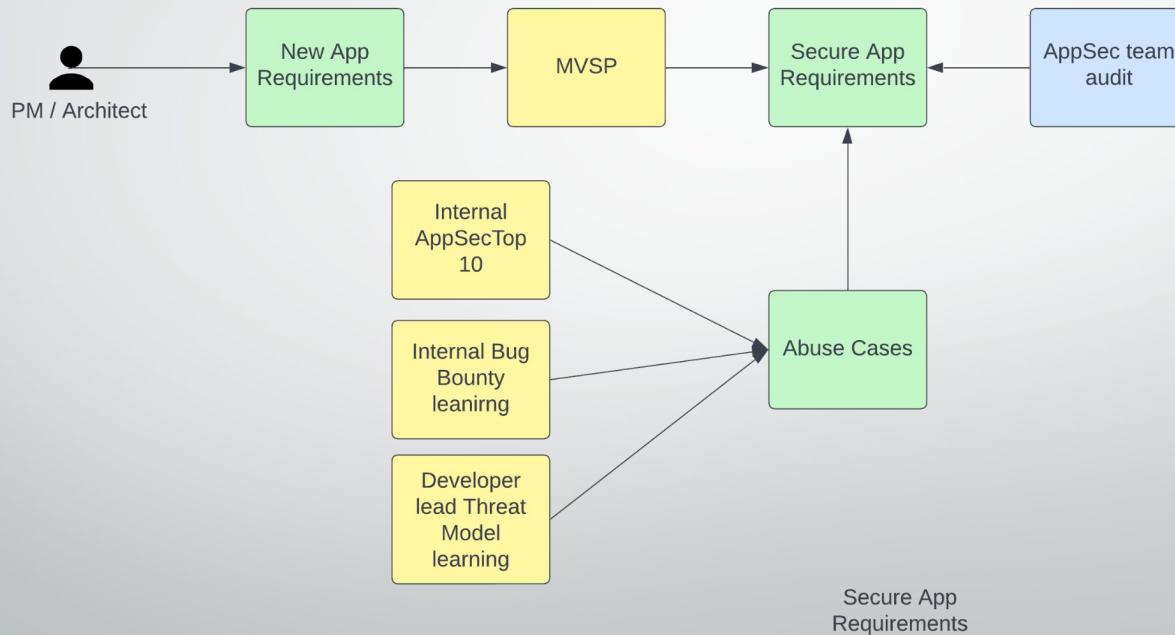
New SAST pipeline

- Minimized security hurdles
- Minimal security tool interaction
- Same or Improved overall risk mitigation
- Reduced load to the Security Team

Patch (SCA/Container/Infra) pipelines



Secure App requirements





Recap



- Define applicable threat actors and acceptable risk profile
- Start small and scale right
- Educate, document and delegate
- Define exception workflow for each AppSec practice
- Monitor each AppSec practice effectiveness
- Update acceptable risk profile based on evolving threats



Thank You!!!



Questions ?