Git Secure Interview Debrief: Template

Interview Script:

Interviewees: Laura Luan and Christophe Elek
Project: *Git-Secure*
Date: *N/A*
Interviewer(s): *Napoleon Santana*
Notetaker(s): *Christopher Geier, John Truskowski*
problem statement: How might we **automate** secure dependency management that will provide the development team with the **notifications and insights** needed to address insecure dependencies on IBM Cloud?

New GitSecure Name: Code risk advisor!

Intro:

Introduce team

Ask if it is ok to record

Introduce our problem statement

Kickoff

How long have you been at IBM?

What projects have you been working on Professionally/Personally?

What do you think about security?

Build

What do you enjoy about your project and what do you do not enjoy?

How do you update your team?

How do you do playbacks?

Grand Tour

What is the current workflow for securing your application?

How does your team utilize notifications?

Who is responsible for security and how is it shared?

Are there any particular tools that help you with your current workflow and which tools hurt your workflow?

Is there a human element when working with security? Should there be more or less human intervention with security?

What do you think about automation? Are you trying to automate your current workflow on a particular platform?

How are security issues patched?

Who is responsible for security and How is it shared?

Reflect

If you could change anything about your workflow what would it be?

Wrap Up

Thank them, for having a productive meeting

Is there anything else you would want to share?

Did we miss anything about your story?

Is there anyone else we should talk to?

Timebox:

intro, problem statement, warm up , kickoff, build ~ 10 min

grand tour ~ rest

reflection ~ 5-7 min

wrap up ~ 5min

Questions	Answer
How long have you been at IBM?	24 years Christophe: 25 years

What projects have you been working on Professionally/Personally?	**Previous Work** network streaming for signals, application for digital studio - high bandwidth data optical networks service level agreements, controlling access **Current work** Defining api and access management IBM Cloud Gateway -> API Management Platform Defining API with YAML Cloud Kubernetes Operator Managing source code compliance and security Christophe: Current work CI/CD in public cloud. Tools for enterprise on Cloud Works with tool to push code on public cloud Pipelines (checking out code/QA) and Toolchain (QA tool, testing tool, git server, cmus, and Insights - is build green, exposure to vulnerabilties)

What do you think about security?	Security in IT is pervasive, across different areas Right now working on source code security Also network security, access control Doesn't handle daily operational role Enforcing security on repository access Christophe Core pipeline and toolchain Project manager, director - is project on track? more resources? progressing? Lower level - health of project? are my tests running? where are people assign? is my development velocity progress? No developer wants to do security. Functional testing is what UI doing. "Doesn't know of any developer that wants to make sure input is validated". Feature addition is design and fun. Security is hidden features not something end user sees as a feature.
What do you enjoy about your project and what do you do not enjoy?	Enjoys new / complex systems Gitsecure composed of multiple components Christophe On a complete devops project. Enhancing NodeJS reporter not meeting needs which is fun. Doesn't like working on 3 different projects in a day.
How do you update your team?	Christophe Daily scrums, frequent slack, ZenHub for clarity on what we are working on. Weekly sprints. Ad hoc meetings for blockers
How do you do playbacks?	

What is the current workflow for securing your application?	**Policy flow to have complete security toolkit for users** **APIs to have different components (7) to leverage other components** **Have complete information about users source code** **Small individual projects -> Larger toolkit for users** **Licence scanning, dependency scanning** **Pipeline triggers for gitsecure scan,** Christophe First process is education. Tools during compile and build steps - white box tools analyzes source code. GitSecure analyzes code in container assumes software is built with some security in mind.
How does your team utilize notifications?	**e.x. HTTPS not implemented on HTTP server. Developer evaluates report and see if this is valid. In this example it was not valid because it was an internal cluster connection.** **e.x. Base images frequent source of vulnerabilities in old images. Could even become vulnerable in 2 months** Christophe Eating your dog food drinking your own champagne Code scanner is run when pushed to development which posts to Slack. "You should not merge from dev to prod" uses a nonblocking approach. **Management looks at this** and depending on severity of CVE alarm sends "OR ELSE" notification. "We can't push to production for vulnerability of highest severity."
Who is responsible for security and how is it shared?	Laura (developer) responsible for resolving responsibilities for applications she wrote Christophe Everyone responsible for security as a software engineering practice. Doesn't have a sole member of team for testing and development security.

Are there any particular tools that help you with your current workflow and which tools hurt your workflow?	**IBM Cloud insights console shows vulnerabilities** **Static scan scans source code line by line for vulnerabilities** Christophe HCL AppScan on Cloud works on Node and Java From development prospective: Golang and transparent compiler with messages at compile time. Jenkins, Github, Git - doesn't like git IDE development and using Eclipse
Is there a human element when working with security? Should there be more or less human intervention with security?	Access controls are frequent Even interviewee doesn't have access to dashboard and needs to request access Its more security, but more bureaucracy to navigate Christophe Of course less security with more human intervention. "The problem of security is human"

What do you think about automation? Are you trying to automate your current workflow on a particular platform?	**Pain point is that latest secure version isn't obvious. Is very annoying that vulnerability is detected but doesn't present direct steps forward on how to resolve. For example providing suggestions on valid version available.** Not much automation for security currently, more automation for testing, etc. Hard to automate for security Checkout and checkin credential for password security, who used which id at what time for enterprise customers requiring audit report. Christophe Loves automation The whole goal of devops checkin of code to production. Move onto next issue after checkin and automate the process after pushing code The number of vulnerability you can test for every second, and now exposure testing can happen instantly. New CVE's tracked in source code and shifting left to verify CVE is fixed automatically. Also need to verify that code running in production is still security, "do you test production for security".
How are security issues patched?	Patch management getting better, automated patch management does upgrade. Cloud deployments makes patching problems go away
Who is responsible for security and How is it shared?	

If you could change anything about your workflow what would it be?	Integration of complex applications and having security requirements. Access issues of not having right credential or losing connections. Simple way to say "enable security with these policies." Focus on application and then security is configured through policies. Connection, credentials, or code? Christophe Test Security, Performance, Features in production Carve out a couple pods for that Continuous testing little by little opening pods up for new customers Some systems not exactly production, VPN, firewall, connection. "You want to fail fast"
Is there anything else you would want to share?	
Did we miss anything about your story?	
Is there anyone else we should talk to?	

Key Takeaways *Overview*	**Tools** *What works and what doesn't*
•	•
Unexpected Information *Surprises*	**Compare to Other Interviewees** *Relative Connections*
•	•

Map of Workflow (Rough Sketch)

Christophe:
Get issue -> Write code -> Test code -> Commit back

Zero day patching workflow
Bill of materials -> CVE detected -> developer notified -> exact code running in production pulled -> patch developed -> push back out to production

Quotes / Conversation	**Interpretations** *Notes/Questions*
Background	
Routine	
Tools	

Challenges	
What was liked in the various software	
Things that would make life easy according to the wish question	