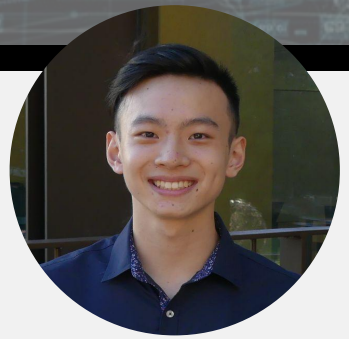
MACQUARIE {wit} ACE ADVISORY

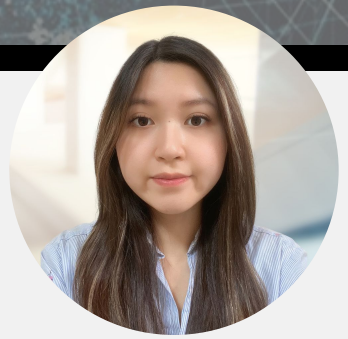
ACE Advisory presents:

AMPLIFICATION

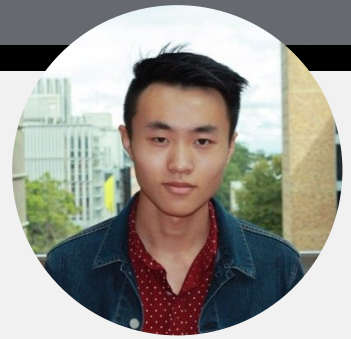
Providing security during and beyond your software development life cycle while maximising the potential of open source



Leo Shi



Truc Luong



Christian Joel

Executive Summary

Issues

- Lack of maintenance despite exponentially growing reliance on open source software
- Exploitation of security loopholes via out of date blackbox third party libraries
- Continually evolving cyber security threats amidst insufficient security procedures

Problem

How can Macquarie leverage the open source community whilst protecting the privacy of their stakeholders from today's evolving cyber-security challenges?

Strategy

Reliability & Vulnerability Scanning CI/CD Pipeline Integration

Macquarie Open Source Sponsorship Hub

Impact

160+
hours saved on software debugging annually

≈ 375
open source software optimisation & maintenance tasks completed by 2025

≈ 70 M
saved by 2025 due to Sysdig Pipelines automating processes

While enterprises are **increasingly relying** on open source for their software development, its **lack of maintenance and optimisation** can lead to **alarming security loopholes**

Out of date libraries cause sourcing inefficiencies

Outdated

54% of respondents don't always check library licences to ensure legal legitimacy

- Potential to introduce bugs
- Affect Macquarie's production time

▼

- Delay analysis and impact advisory

Macquarie's technology offerings deal with millions of dollars

Digital Banking Offering

Global Commodities

How can Macquarie leverage the open source community whilst protecting the privacy of their stakeholders from today's evolving cyber-security challenges?

Slow Testing

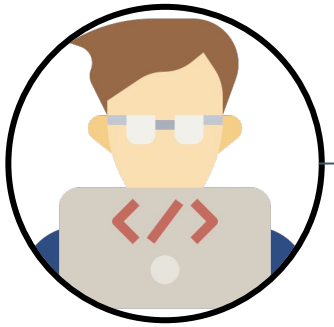
Lack of understanding can cause flaws to take up to 7+ months to fix

- With proper code knowledge, 42% of flaws are fixed within one week
- Slow correction of bugs damages brand image
- Contradicts Macquarie's security mantras

Data breaches could have dramatic ramifications

- Jeopardise the privacy of sensitive data, intellectual property and personal information
- Human error accounts for 30% of financial data breaches

The rise in **cyber security threats** alongside the **growing reliance** on **open source software** serves as a challenge for both Macquarie and the open source community

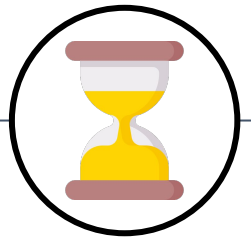


Archit
Macquarie
Developer

Archit found an **open source library** that met all his needs for the project

The project was successfully **deployed on time** for the Financial Services team

4 months later, the Financial Services team reported a **data breach** from their **client**



This breach came from the open source library and would take **months to resolve**

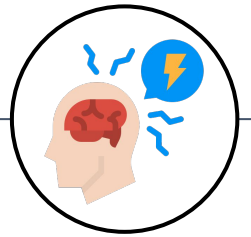
Undetected security issue which costs **money, reputation**, and even **legal issues**

Suz
Open Source
Developer

Suz publishes her programs on GitHub to help others with their programming needs

She enjoys taking **control of her working hours** and contributing to the **open source community**

Unfortunately, the **lack of income** means that she cannot treat this as her main job



She **discontinued** the maintenance of her open source programs as it is **impossible to manage** with her job

Lack of income leading to **discontinued** though **useful** open source software

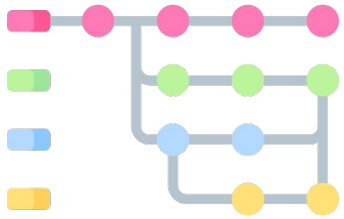
Analysis | **Strategy** | Implementation | Impact

MACQUARIE

By integrating a **reliability check script** as a pipeline job, unexpected errors and bugs can be detected and resolved at the development stage

Code → **Build**

Leverage desired third party libraries and other open source software without any concerns



Reliability check job is automatically triggered with every code commit flagged as open source, scheduled via GitLab

Reliability Check Procedure

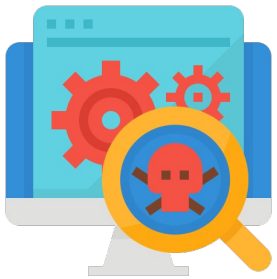
1. Search for origin of the open source software used
2. Retrieve and analyse reviews and feedback from open source communities (e.g. AlienVault OTX) about the software
3. Return list of potential errors and bugs associated to the software and their fixing recommendations
4. Job only passes if list is empty, or if all errors and bugs are marked as addressed

Reduce unnecessary **trouble shooting** for developers

Enhanced **reliability** from development stage

The implementation of **Sysdig Secure** (integrated security tool) provides protection at every stage of Macquarie's development cycle in a streamlined manner

Build | **Deploy**



- Sysdig Secure **vulnerability scanning** job is automatically triggered with every code commit
- Developer is notified of job failure for **immediate security fix**

- Automated runtime **threat detection** and **incident response** even after deployment
- **Network** security
- Consistent **visibility** of security via **visualisations** and **alerts**

Why Sysdig Secure?

1. **Immediate integration** with Macquarie's existing DevOps architecture
2. Protection at **every stage** of the software development life cycle
3. Accelerates Macquarie's **automation** and **cloud migration** journey with **containerisation**

Macquarie's Open Source Sponsorship Hub will **ensure** open source developers receive **substantial monetary benefits**

Open Source Maintenance/Debugging/ Optimisation Task Development

1 Task identified by relevant development team

2 Task **effort, deadline, payment** and **description** determined by development team (Jira ticket)

3 Provide public Git project link with dummy data

Task is published on the **Sponsorship Hub**

Task Picked Up by Open Source (OS) Developer

4 Assign themselves to task(s) which they find suitable

5 Work directly on provided public Git project link

6 Project automatically submitted at deadline

Submitted project fed into **DEV environment**

Quality Testing & Verification

7 Submitted project tested for **security, reliability and accuracy** using real project's development/lab environment pipeline

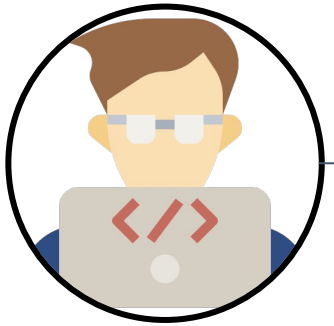
Payment sent as per **4 component checklist** (quality, time, complexity and importance)

Optimised and **future proofed** black-boxed open source software frequently utilised by Macquarie

Open Source Developers gain **monetary** benefits while maintaining **freedom**

Maximise the potential of open source software in a **secure** manner

Amplification empowers Macquarie and open source developers to maximise their potential while placing cyber security as core focus



Archit
Macquarie
Developer

The automated reliability and security checks help Archit **identify** and **fix** bugs that were not covered by the test suite

Archit can now fix these errors within **one working day** using the provided recommendations

Even after deployment, Archit still has **full visibility** of the **security** of the project via Sysdig's friendly GUI

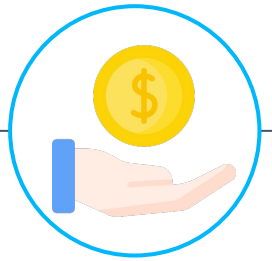
Security alerts are **automatically forwarded** to the security team without needing Archit to submit a review ticket

Archit's team can now push out new developments **faster** while maintaining **quality** and **security**

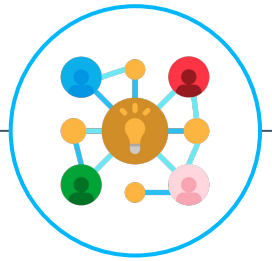
Suz
Open Source
Developer

On the task hub, Suz can **choose** which job she would like to take on

She is once again **in control** of her working hours and schedule



After her submission passes Macquarie's review check, Suz can immediately receive **payment**



She feels **connected** and **supported** by Macquarie and other developers in the hub

Suz can continue maintaining her other open source projects while having a **sustainable income**

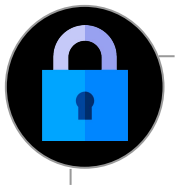
Analysis | Strategy | Implementation | **Impact**

MACQUARIE

The impacts of **Amplification** are **mutually beneficial** for Macquarie, their clients and the stimulated open-source community

Cybersecurity enhanced for company and client benefit

Cycle of **Vulnerability Management** established that allows for immediate identification & remediation of software vulnerabilities



Automated & scheduled security checks and incident responses allows for **Configuring and Hardening** of MQ systems

Attestation and Security Concerns are upheld to certificate standards via Sysdig Secure's frequent updates in accordance to international cyber security standards

Financially incentivising the open-source community

An average of **$1025** rewarded per solved issue!

Still retaining the **freedom** of an open-source developer

Allowing Macquarie's programs to remain **cutting edge** by **promoting the open-source community** whilst securing all company and client information behind **constantly fortified** cybersecurity

ACE ADVISORY

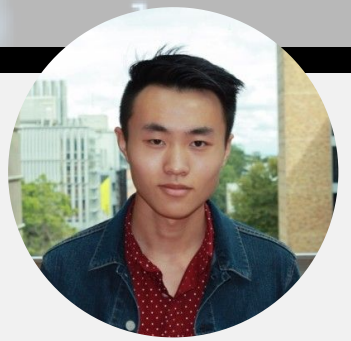
MACQUARIE

{wit}

Q&A

Leo Shi

Truc Luong



Christian Joel

Appendix Network

Main

1. Executive Summary
2. Problem Analysis
3. User Journey (pre-solution)
4. Reliability & Vulnerability Scanning CI/CD Pipeline Integration
5. Why SysDig?
6. Macquarie's Open Source Sponsorship Hub Process
7. User Journey (post-solution)
8. Impacts

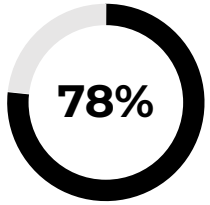
Appendix

1. Open source is the future
2. Macquarie's need for cyber security
3. Case Studies: Heartbleed, Canva's Data Breach
4. Definitions: Containerisation, Kubernetes, Monolithic vs Microservices
5. Strategy Overview
6. GitLab Pipeline with Reliability Check
7. Macquarie's Development Practices
8. Popular Recommendations for Security Enhancement & Mitigation of Open Source Risks
9. Other OSS Security Tools
10. Falco vs Sysdig Secure Comparison
11. Sysdig Secure Background Development
12. Sysdig Secure Pipeline Integration
13. Sysdig Secure GUI Examples
14. Sysdig Secure across the Development Lifecycle
15. How does Sysdig Secure's vulnerability scanning work?
16. How does Sysdig Secure detect and respond to runtime threats?
17. How does Sysdig Secure meet cyber security standards?
18. How can Sysdig Secure support Macquarie's containerisation journey?
19. Sysdig Secure Available Integrations & Current Users
20. Sysdig Secure Service & Support Packages
21. Macquarie Open Source Sponsorship Hub in General
22. Macquarie Open Source Sponsorship Hub's Interface: (1), (2)
23. Macquarie's 4-component checklist to categorise problem funding
24. **FAQ:** Macquarie is paying developers to optimise their open source software, but other people also benefit from this. Does this seem fair?
25. Implementation
26. Financials: Financials Summation, Expenses, Cost Savings Calculations
27. Risk & Mitigations

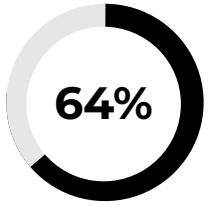
MACQUARIE

Open source is **consistently growing**, but its **maintenance** still needs to be addressed

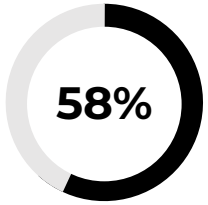
Statistics suggest open-source is rising in popularity



Companies run on open source - less than 3% don't use OSS



Companies participate in open source projects



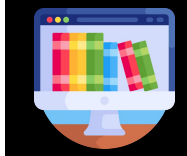
Companies believe open source scales better

Yet, open-source is still not a viable career path in 2022

- Developers view open-source as a side hobby
- Becoming a full-time open-source developer is unsustainable financially
- This leads to a lack of accountability for maintenance

Source: ZDNet (2015), CPO Magazine (2021)

Out of date libraries cause sourcing inefficiencies



Outdated Licences

54% of respondents don't always check library licences to ensure legal legitimacy

- Potential to introduce bugs
- Affect Macquarie's production time

- Delay analysis and impact advisory streams
- Diminishes customer satisfaction

Slow Testing

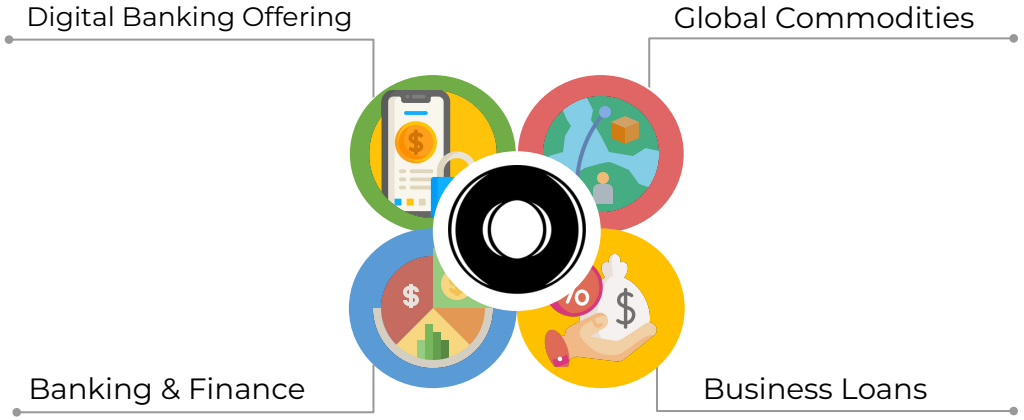
Lack of understanding can cause flaws to take up to 7+ months to fix

- With proper code knowledge, 42% of flaws are fixed within one week

- Slow correction of bugs damages brand image
- Contradicts Macquarie's security mantras

Macquarie needs a robust cyber security system to protect the **high amounts of personal information** the company handles on the daily

Macquarie's technology offerings deal with millions of dollars



Data breaches could have dramatic ramifications

- Jeopardise the privacy of sensitive data, intellectual property and personal information
- Human error accounts for 30% of financial data breaches

Cybercrime rates are growing and evolving every day

- Costed the world $6 trillion in 2021
- Potential for passwords, credit card details & tax file numbers to be leaked
- Macquarie needs robust security to prevent human error

3 aspects of cyber security

1. Vulnerability management
2. Secure configuration and hardening
3. Attestation and security certifications

How can Macquarie leverage the open source community whilst protecting the privacy of their stakeholders from today's evolving cyber-security challenges?

Source: The Adviser (2021), IBM (2021)

Heartbleed

Heartbleed showed us that we cannot blindly trust open-source software

- The Heartbleed Bug was introduced into the software in 2012 and publicly disclosed in April 2014 - 2 whole years without detection
  - **Improper input validation** in the OpenSSL implementation of the **TLS Heartbeat extension**
  - Attackers can send Heartbeat requests with the value of the length field greater than the actual length of the payload.
    - Heartbeat requests don't verify if the payload size is same as what is specified in length field. Thus, the machine copies extra data residing in memory after the payload into the response. Therefore, the extra bytes are additional data in the remote process's memory.

- The Heartbleed Bug is a serious vulnerability in the popular **OpenSSL** cryptographic software library.
  - The Heartbleed bug allows anyone on the Internet to read the memory of the systems protected by the vulnerable versions of the OpenSSL software.
- This compromises the **secret keys used** to identify the service providers and to encrypt the traffic, **the names and passwords of the users** and the actual content.

Sources:
https://heartbleed.com/
https://www.synopsys.com/blogs/software-security/heartbleed-bug/#:~:text=The%20Heartbleed%20bug%20results%20from,was%20first%20discovered%20in%202014.

MACQUARIE

Canva’s Data Breach

Canva’s continuous investment into cyber security after data breach

- Canva’s systems were breached on Friday May 24 of 2019 and "up to" **139 million users**’ details - comprising usernames, email addresses and hashed passwords - were stolen.
  - Occurred on Friday because all major security incidents begin as you’re going into the weekend

- 2 years on from this (2021), Canva is “still growing” their security team and investing in cyber security to prevent this from happening again
- The incident had “influenced the culture at Canva”, resulting in more resourcing and investment being put behind security.
  - The event from two years ago had a really visceral impact on company executives
- Outlines how **traumatic it is if a company** was to go through a data breach

Source: https://www.itnews.com.au/news/canvas-infosec-resourcing-still-growing-two-years-after-large-data-breach-569282#:~:text=Canva's%20systems%20were%20breached%20on,%E2%80%9Cattack%20on%20our%20systems%E2%80%9D.

Canva

MACQUARIE

Containerisation

Containerisation is a DevOps Standard

- **Containerisation** is operating system-level virtualisation or application-level virtualisation over multiple network resources so that software applications can run in **isolated user spaces** called containers in any cloud or non-cloud environment, regardless of type or vendor.
  - **Cloud portability** is the selling point: containers typically mean that programmers won't have to **rewrite** the code for each new operating system and cloud platform.
  - If you want to leverage containers from public repositories, you should look for those that are **signed by the Docker Content Trust system** to ensure that you're downloading a legitimate container.

Issues with Containerisation

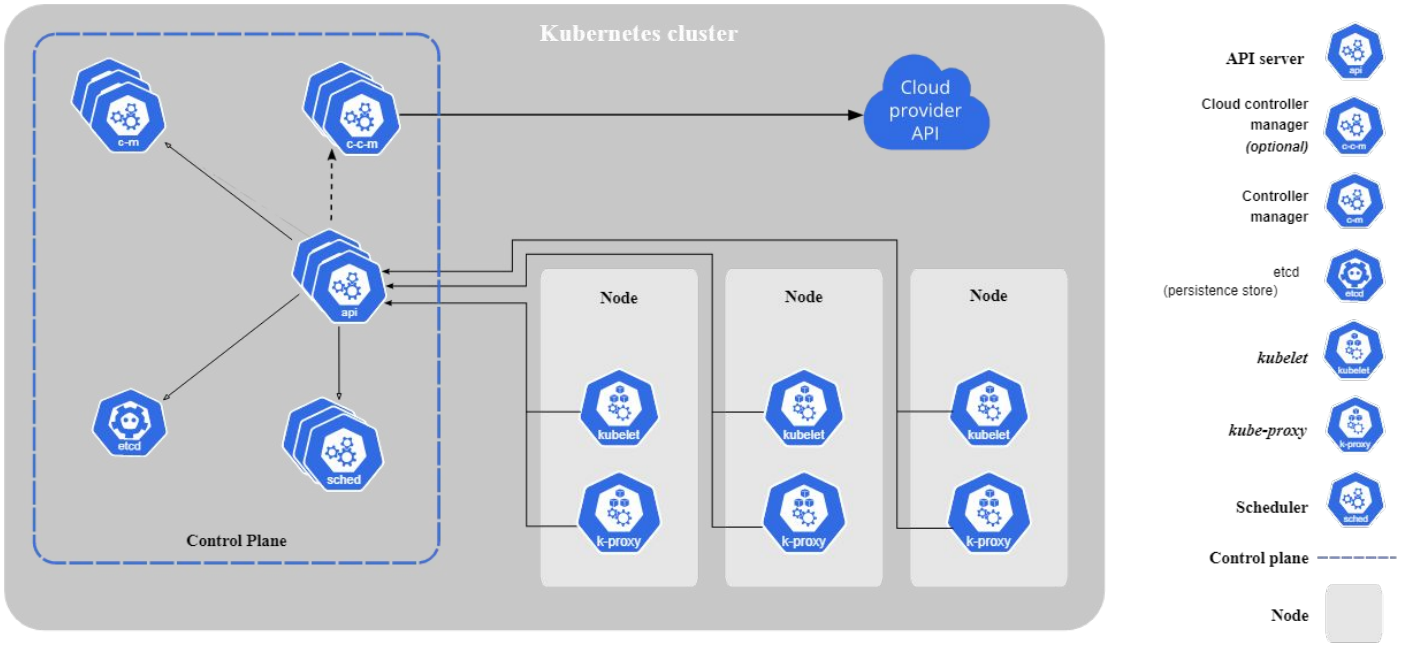
- Lack of knowledge and practice in the industry
- Requires frequent testing

Source:
https://techbeacon.com/enterprise-it/youve-heard-benefits-containers-now-understand-challenges

MACQUARIE

Kubernetes

- Kubernetes is a portable, extensible, open source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation.
- Kubernetes does not limit the types of applications supported
  - Aims to support an extremely diverse variety of workloads



Kubernetes provides you with

- Service discovery and load balancing
- Storage orchestration
- Automated rollouts and rollbacks
- Automatic bin packing
- Self-healing
- Secret and configuration management

MACQUARIE

Monolithic vs Microservices

Monolithic Application

Deployed on a set of identical servers behind a load balancer

The traditional way of building applications

- Built as a single and indivisible unit

Strengths

- Less cross-cutting concerns
- Easier debugging and testing
- Simple to deploy and develop

Weaknesses

- Scaling up can become too complicated to understand
  - can only scale up the whole application
- Hard to implement changes

Microservices

Consists of a large number of services, each with multiple runtime instances

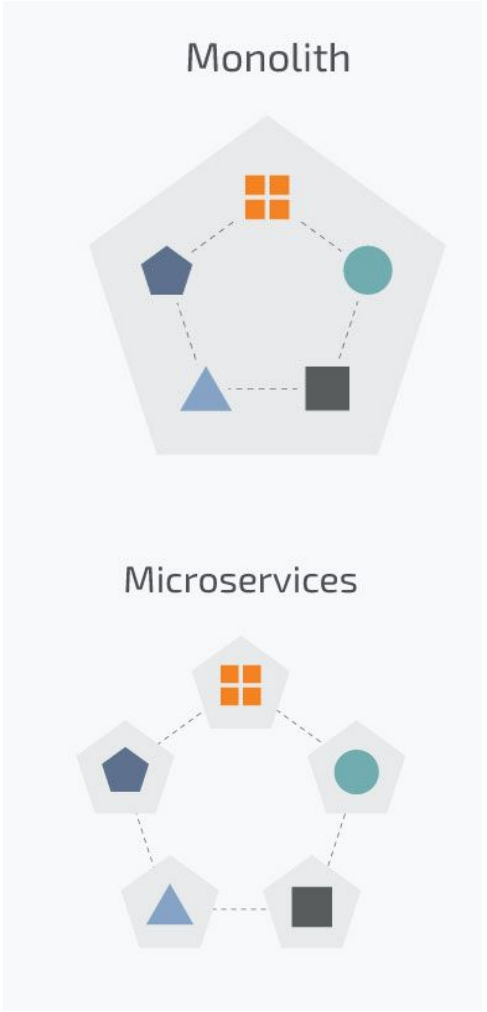
A collection of smaller independent units

Strengths

- Independent components
- Easier understanding stemming from the ability to split applications
- Better scalability

Weaknesses

- Extra initial complexity
- Testing required to analyse cross compatibility
- More complicated system distribution



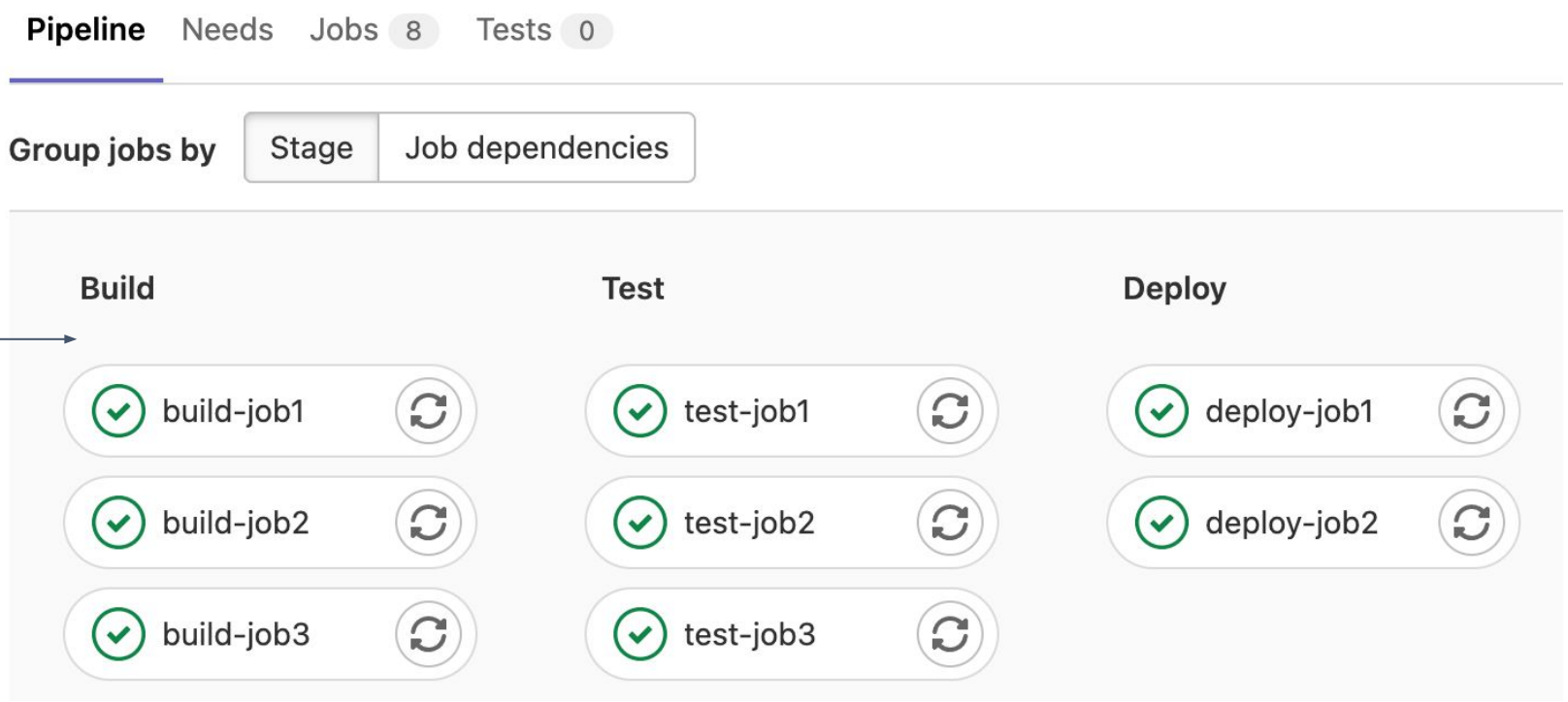
MACQUARIE

Strategy Overview

Strategy	Execution	Impact
 Reliability & Vulnerability Scanning Pipeline Integration with **Sysdig Secure**	**Reliability** evaluated via open source community reviews to flag bugs and security issues	• Improve **performance, reliability and security** of deployed programs and features
	Vulnerability scanning integrated into CI/CD pipeline, with automated alerts and incident response	• Establish Macquarie's position as a **prominent enterprise supporter** in the **open source community**, and l**eader in cyber security**
 Macquarie Open Source Sponsorship Hub	**Paid open source software** upgrade and optimisation opportunities for volunteer developers centralised in one public developer hub	• Accelerate Macquarie's ability to achieve its **automation** and **cloud migration** goal

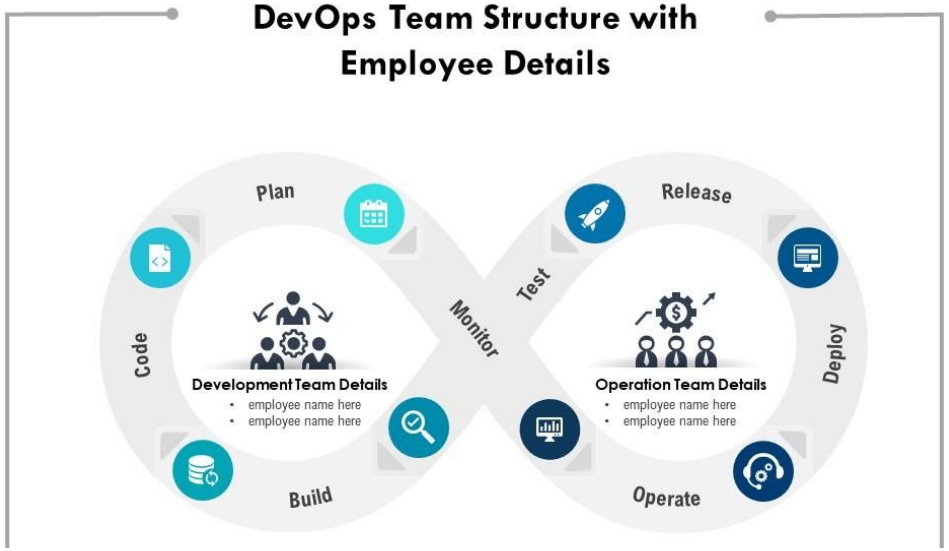
GitLab Pipeline with Reliability Check

Reliability Check
Build Job (Script)



Macquarie’s Development Practices

- Cloud first approach for the whole group’s infrastructure
- One of the north stars was revealed by BFS CTO Jason O’Connell at a Google Cloud summit in September, when he suggested Macquarie wouldn’t stop at having 100 percent of its workloads on infrastructure-as-a-service, but would instead chase a **fully cloud-native environment** on software-as-a-service (SaaS) or platform-as-a-service (PaaS) → trying to achieve **by 2025**!
- “An example of one of our north stars is we actually feel that we could **automate 100 percent of our testing** or get close to [that].”
- **Strong DevOps culture** - The core DevOps principles at Macquarie BFS are: to own things end-to-end and break down silos; that failure happens and the key is to detect, recover and adapt fast; implement small changes frequently; automate routine tasks; measure everything, and; to share knowledge and be open to learning.
- **Containerisation** is operating system-level virtualisation or application-level virtualisation over multiple network resources so that software applications can run in isolated user spaces called containers in any cloud or non-cloud environment, regardless of type or vendor.
- **Cloud portability** is the selling point: containers typically mean that programmers won't have to rewrite the code for each new operating system and cloud platform.
- Kubernetes helps you to **build cloud-native microservices-based apps**. It also supports **containerisation of existing apps**, thereby becoming the foundation of application modernisation and letting you develop apps faster.



Manual efforts to monitor the introduction of new OSS struggle to keep up with the pace of code churn in a DevOps environment, where dozens of new releases can be seen each day. Rapid changes to an application can result in a rapidly changing risk profile. However, stopping builds and slowing down pipelines to review changes manually would be an impractical “gating” practice—going against the natural development flow of DevOps and frustrating teams.

Source: https://www.itnews.com.au/news/macquarie-group-embraces-secure-by-design-569489

MACQUARIE

Popular Recommendations for Security Enhancement & Mitigation of Open Source Risks

- DevSec teams to i**ntegrate security earlier in SDLC** and integrate open source software securely from the start
- **Automation tools** can provide enormous value for tracking open-source components and their status as well as for evaluating components. Open source code can be scanned before and during use through Dynamic Application Security Testing (DAST) or Static Application Security Testing (SAST) tools.
- Policies should require consideration of an **open-source component's history**, such as the density of known issues, version release frequency, and latency between issue identification and patch
    - Important to know how robust the community involved in a project is and anticipate what sort of support it might or might not provide.
- Policies need to dictate what sources and licence types are acceptable for use and should help developers decide whether to use individual components or an entire codebase.

Backbone of our reliability scanning solution

Ways to mitigate open source risk

To protect against vulnerabilities and malware in open source code, every company must take four specific steps.

- **Create and enforce security policies**. Companies must have policies that govern how developers access and use open source libraries.
- **Understand what open source libraries are being used and where the vulnerabilities are**. Companies can most easily accomplish this through specialized static analysis (see below).
- **Update vulnerable libraries**. Cooperation between app security teams and development teams is critical here, as updates to libraries can sometimes break applications. Libraries may not need to be updated if developers aren't using the vulnerable parts of the code.
- **Mitigate malware**. The most effective a way to stop malware is to create warnings for developers who are accessing vulnerable libraries, and to create enforcement rules on Continuous Integration servers that will fail the build if vulnerable parts of libraries with high open source risk are used within the code.

Source: https://www.veracode.com/security/dangers-open-source-risk

MACQUARIE

Other OSS Security Tools

AlienVault OSSIM	AlienVault USM
• Leverages AlienVault Open Threat Exchange (OTX) by allowing users to both contribute and receive real time information about malicious hosts • Requires manual configuration - at its core, AlienVault only gives you data - requires security team to manage AlienVault services • Powered by AT&T -> has regular updates • Doesn't have log management, cloud infrastructure monitoring, security automation, continuously updated threat information and visualisation • No third party integrations • Single server architecture • On-premise/VM only - no cloud support	• Subscription version of AlienVault OSSIM • Provides log management, cloud infrastructure monitoring, security automation, security automation, continuously updated threat information and visualisation • Single server deployment only • Provides cloud support, including AWS and Azure • Only suitable for small to medium sized companies • Requires manual configuration for pipeline integration

- Wireshark only provides network security tools and has difficult to use GUI (source: https://www.trustradius.com/compare-products/wireshark)
- Kali Linux requires manual on premise configuration for ethical hacking purposes
- Cloud only security tools only provide vulnerability scanning post deployment and not during development - what we need is end to end protection!

MACQUARIE

Falco vs Sysdig Secure Comparison

Falco Open Source Security and Sysdig Secure: Feature Comparison

	Falco	Sysdig Secure
Open Source Based Agent	✓	✓
Runtime Security		
Threat Detection Policies (via Linux syscalls, Kubernetes audit logs and cloud activity logs)	✓	✓
Alert Outputs	✓ (via Sidekick)	✓ (Event forwarding)
Customizable Policies Based Cloud/K8s Context	✓	✓
Automated Policy Tuning	✗	✓
ML-based Image Profiling	✗	✓
Network Security	✗	✓
Additional Capabilities		
Out-of-the-box Compliance Policies	✗	✓
Vulnerability Management (Image & Host scanning)	✗	✓
Cloud Security Posture Management	✗	✓
Infrastructure as Code Security	✗	✓
Incident Response	✗	✓
Enterprise Grade Support and Scalability (centralized rule management, simple policy editor, professional services)	✗	✓

Source: Sysdig

MACQUARIE

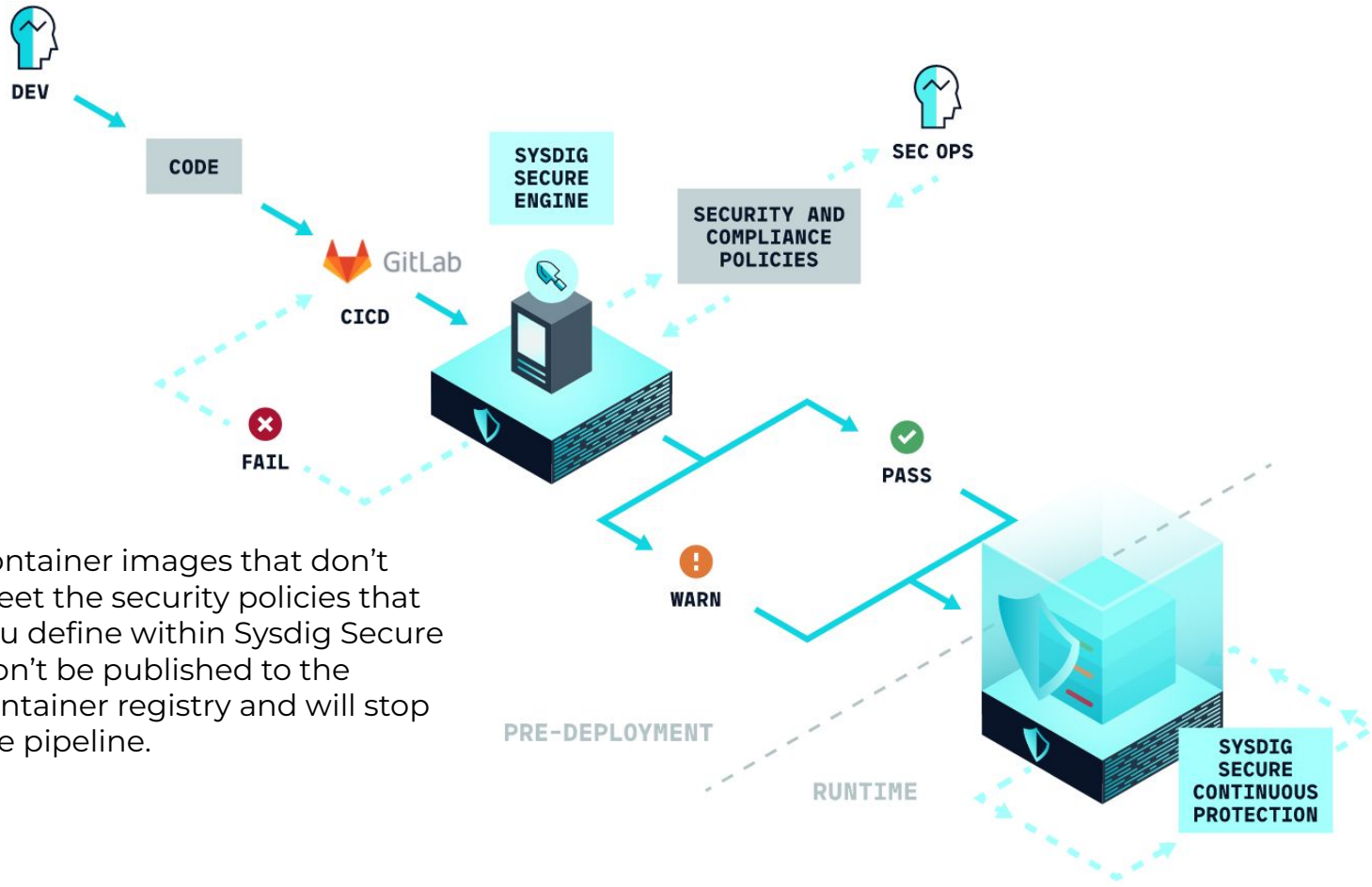
Sysdig Secure Background Development

Based on Open Standards with No Black Boxes

Easily Integrate with other tools you use. Detect anomalies based on Falco, the cloud native standard for threat detection. Enforce consistent policies based on OPA, the cloud native standard for configurations. Maximize coverage with community-sourced detection rules that are easily customizable.

Source: Sysdig

Sysdig Secure Pipeline Integration



Container images that don't meet the security policies that you define within Sysdig Secure won't be published to the container registry and will stop the pipeline.

Source: https://sysdig.com/blog/gitlab-ci-cd-image-scanning/

Sysdig Secure offers a full featured **container image scanning** functionality, among many other container security features like **runtime threat detection** with machine-learning-based profiling and extensive out-of-the-box **detection patterns**, enforcement using Kubernetes PSPs, **incident response** and forensics or compliance.

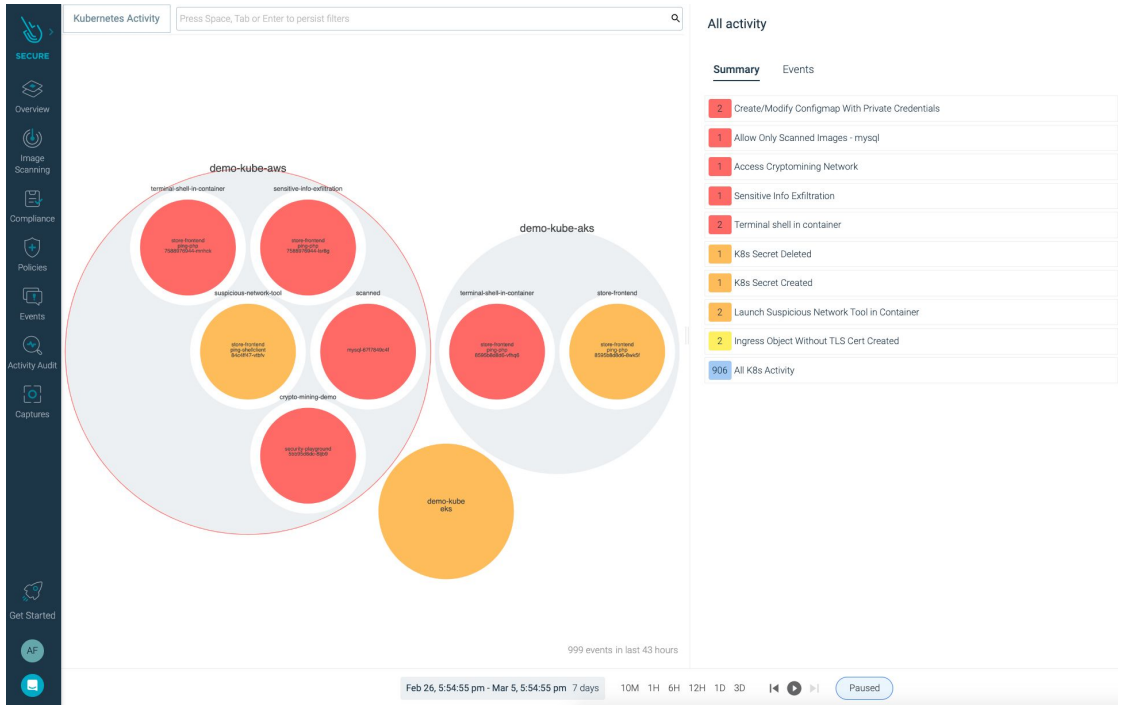
Integration Steps

1. Configure access credentials
   a. GitLab account
   b. Sysdig platform account
2. GitLab pipeline definition to:
   a. Build the container image
   b. Scan the image for vulnerabilities or policy violations
   c. Push the image to the final repository, which can be specified by developer

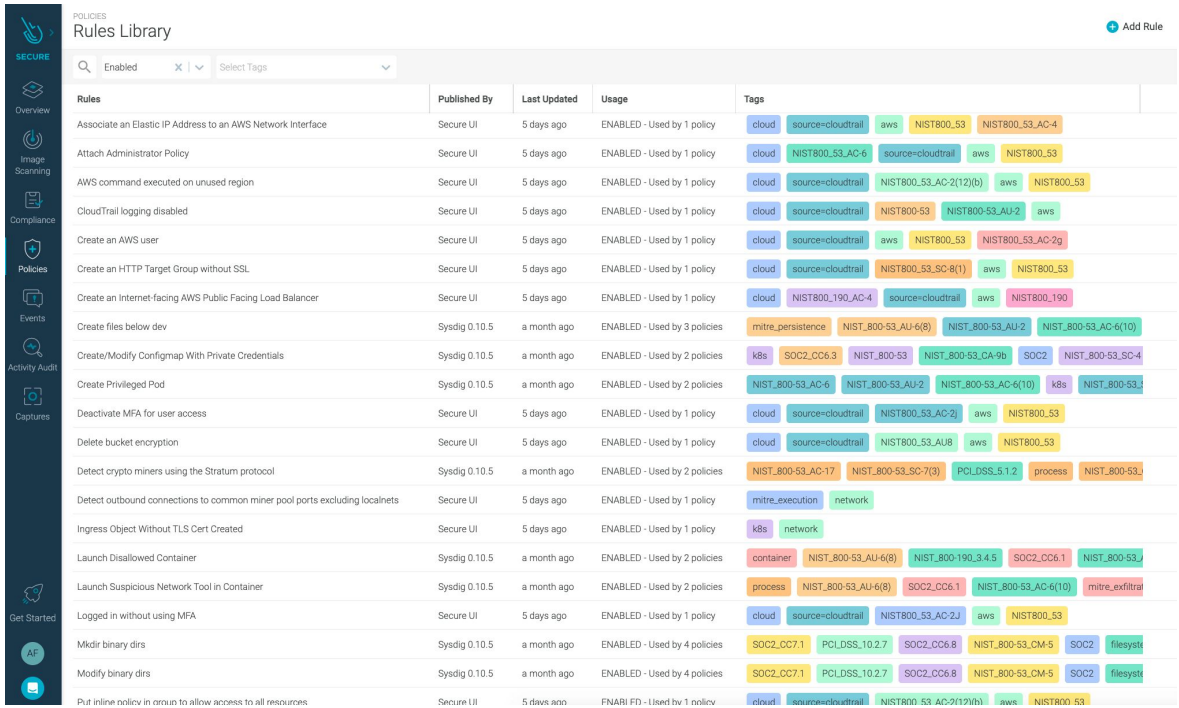
MACQUARIE

Sysdig Secure GUI Examples

Cloud Security Management:



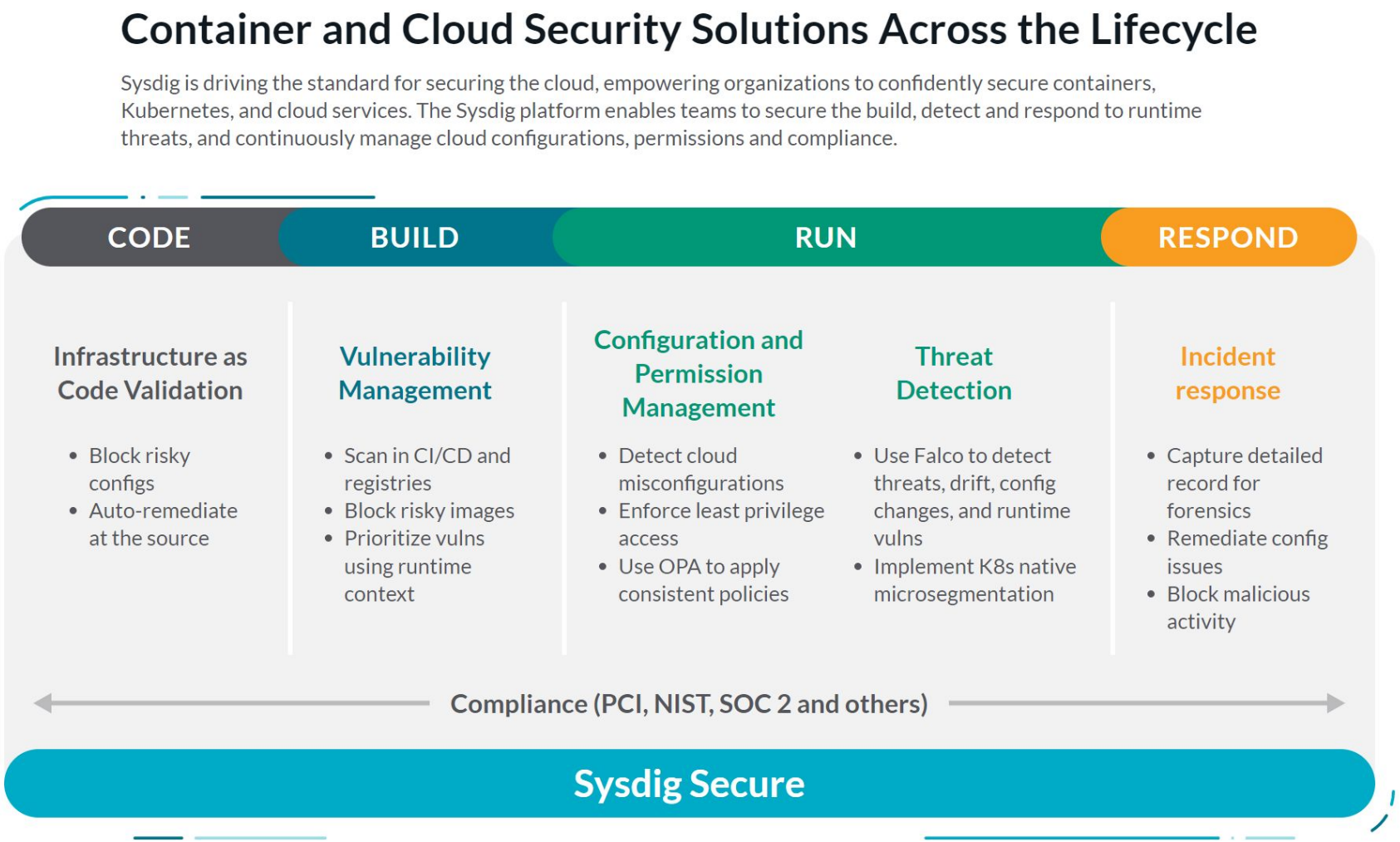
Runtime Security:



Vulnerability Management

MACQUARIE

Sysdig Secure across the Development Lifecycle



Source: Sysdig

MACQUARIE

How does Sysdig Secure's vulnerability scanning work?

Identify Container Vulnerabilities Pre-Production and at Runtime

Automate image scanning within CI/CD

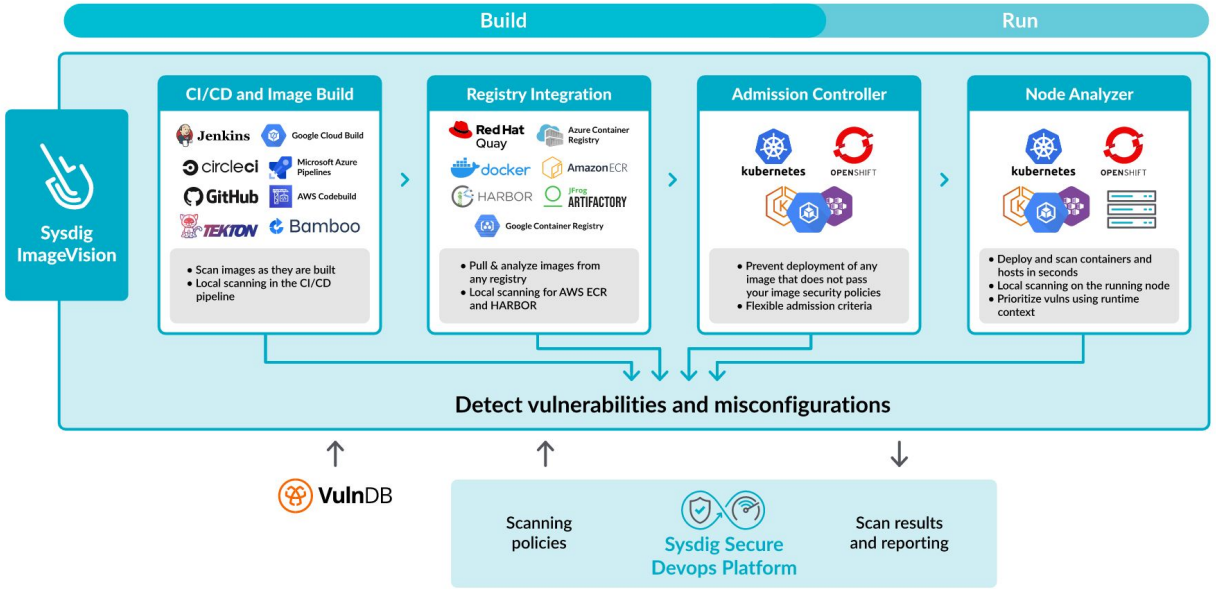
Detect OS and non-OS vulnerabilities early by embedding image scanning (docker security scanning) tools into CI/CD and registry scanning before deploying to production.

Single vulnerability management solution for containers and hosts

Save time and money by consolidating host and container vulnerability scanning in a single workflow. Deploy and scan in seconds.

Prioritize vulnerabilities with runtime context

Continuously detect and automatically prioritize vulnerabilities using runtime context. Eliminate noise, stop vulnerability overload, and fix what is important faster.



5 KEYS TO A SECURE DEVOPS WORKFLOW

Manage Vulnerabilities: Scan container images and hosts

As the number of container images, versions, and builds proliferates, you lose control of what software is being used and whether software updates are applied. Embedding security into your delivery pipeline as you build applications helps you identify and address vulnerabilities faster, and keeps your developers productive. Here are steps you can take to get control.

Embed scanning into CI/CD pipelines and registries to prevent risky images from being deployed.

Validate the build configuration (Dockerfile instructions) and image attributes (like size and labels).

Alert the right team for each issue (notify the owner of each image and integrate with your CI/CD tool to show the scan results directly in that context).

Adopt in-line scanning and maintain full control of your images.

Create different policies for each workflow, including images from public repositories and images built in-house. Consider different checks for each app.

Automatically scan physical, virtual and cloud based host instances to identify vulnerabilities in OS and non-OS packages.

Identify new vulnerabilities that impact the image once the container has been deployed.

By integrating security analysis and compliance validation into this process, you can address issues earlier so you don't slow down deployment. This is known as "shifting security left."

Source: Sysdig

MACQUARIE

How does Sysdig Secure detect and respond to runtime threats?

Leverage Kubernetes-native controls for runtime protection of cloud-native workloads.

→ Use Admission Controllers to allow or block specific configurations and determine whether the container can be run on the cluster.

→ Prevent attacks by enforcing “least privilege” on containers through Pod Security Policy (PSP). PSPs control what permissions pods get at runtime (e.g., which user is running in privileged mode, whether they have access to the host network or filesystem, etc.).

Monitoring CPU and other resource usage is relevant for security, as they are typically exploited in DoS and crypto-mining attacks. Monitoring network connections gives you information about the attack, runtime behavior, and spread vectors. Some attacks are first detected as monitoring alerts rather than security violations.

Create and maintain a runtime policy that observes workload behavior, cloud activity, and identifies anomalous events.

→ Leverage tools to automatically build and customize policies or use out of the box Falco rules. → Upgrade of Falco

→ Implement least-privilege and compliant network policies with K8s and app metadata.

→ Simulate the effects of runtime policies before applying them in production to avoid breaking application functionality.

→ Visualize network communication in and out of a particular pod/service/app/tag over time with topology maps. → Provides visualisation for security team

→ Apply the right security policy based on container role and Kubernetes context.

→ Automate use of events in cloud logs to detect threats and configuration changes on cloud services. → In line with Macquarie’s automation vision

Streamline incident response and quickly respond to container and cloud security threats with a detailed activity record. Use capture files based on syscall data enriched with Kubernetes and cloud context to quickly answer the questions of “when”, “what”, “who” and “why” for your container security incidents. This detailed record allows you to conduct post-mortem analysis and determine root cause, even after containers are gone.

Source: Sysdig

MACQUARIE

How does Sysdig Secure meet cyber security standards?

Check your container and platform configuration against CIS benchmarks for Docker and Kubernetes.

Validate compliance during the build, mapping container image scanning policies to standards (e.g., NIST, PCI, SOC2, or HIPAA) or internal compliance policies (e.g., blacklisted images, packages, or licenses).

Implement File integrity Monitoring (FIM) to detect tampering of critical system files, directories, and unauthorized changes. FIM is a core regulatory requirement for a number of compliance standards.

Manage compliance at runtime. Check for best practices (e.g., don't run privileged containers and don't run containers as root) and look for known adversary tactics and techniques. Achieve and maintain compliance with security frameworks mapping through a rich set of Falco rules for security standards and benchmarks, like NIST 800-53, PCI DSS, SOC 2, MITRE ATT&CK®, CIS AWS, and AWS Foundational Security Best Practices.

Provide proof of compliance with capture files that incorporate detailed forensics data and host scanning reports. It's important to record configuration and policy changes, including an audit of runtime changes for compliance audits.

Implement compliance checks to meet regulatory compliance standards (CIS, SOC2, PCI, NIST 800-53, etc.) across containers, hosts, Kubernetes, and cloud. Monitor cloud services continually for configuration drift that can impact compliance. Measure compliance progress with scheduled assessments and detailed reports.

Source: Sysdig

How can Sysdig Secure support Macquarie's containerisation journey?

Monitoring the dynamic nature of container-based applications is critical for the high availability and performance of cloud services. Microservices-based applications can be distributed across multiple instances, and containers can move across multi-cloud infrastructure. Monitoring the Kubernetes orchestration state is crucial to understanding if Kubernetes is keeping all of the service instances running.

- → Monitor health and performance with deep visibility into infrastructure, services, and applications. Get the operational status of your cluster with Kubernetes orchestration monitoring.
- → Immediately identify owners for issue resolution using container and cloud context.
- → Identify pods consuming excessive resources and monitor capacity limits. Control unexpected billing and application rollouts and rollbacks of deployment by monitoring auto-scaling behavior.
- → Reduce cost by optimizing capacity across clusters and cloud.

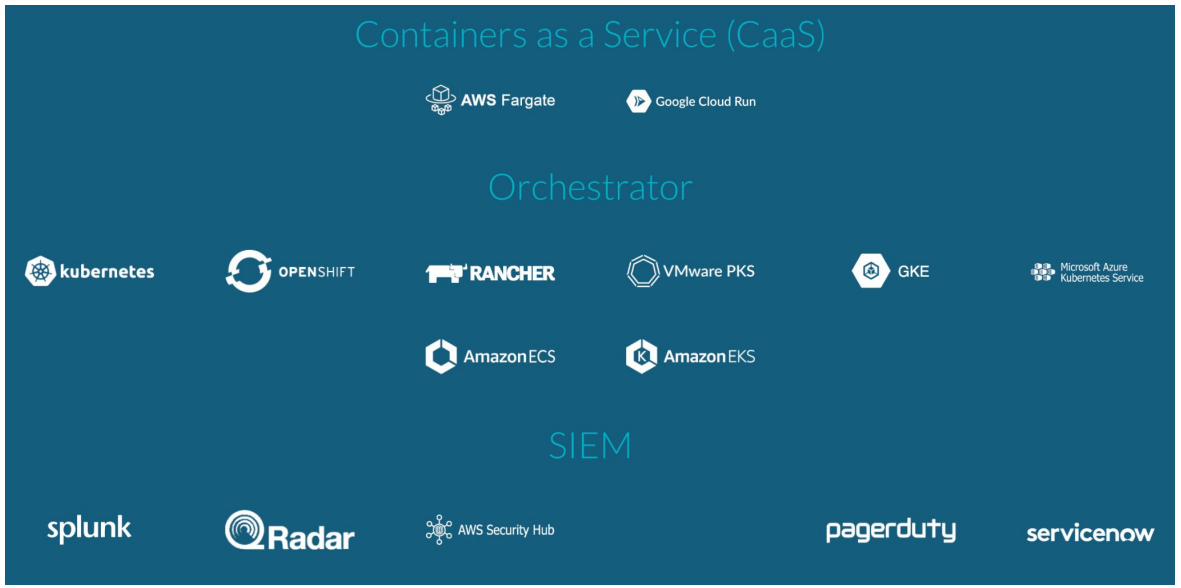
Improve application performance and rapidly solve issues with deep container visibility and granular metrics enriched with Kubernetes and cloud context. You can monitor the impact of a given security incident on service availability.

Get productive quickly by using Promcat.io, a resource catalog of Prometheus integrations with curated, documented, and supported monitoring integrations for Kubernetes platform and cloud-native services.

Source: Sysdig

MACQUARIE

Sysdig Secure Available Integrations & Current Users



MACQUARIE

Sysdig Secure Service & Support Packages

Service and Support

	Standard	Premium^^
Cost	Included in agent price	15% of total product cost
Mon. - Fri. Technical Support	✓	
24/7 Technical Support		✓
Email Support	✓	✓
Knowledge Base	✓	✓
Phone Support		✓
Severity & Response Times	P1 - 4 Hours SLA P2 - 8 Hours SLA P3 - Next Business Day	P1 - 30 Minutes SLA P2 - 2 Hours SLA P3 - 4 Hours SLA P4 - Next Business Day

^^Required for On-Prem Deployments

Source: Sysdig

MACQUARIE

Macquarie's Open Source Sponsorship Hub will **ensure** open source developers receive **substantial monetary benefits**

We will build the Sponsorship Hub with an easy to use UI to minimise developer's time spent looking for jobs

- Intuitive **Jira-like** interface for ease of transition
- **Clear** documentation for projects
- Funding determined through a **4-component checklist**
  Based on **quality** of prescribed code, assumed **time** needed, problem **complexity** & **importance** of task

By leveraging the open source community, Macquarie can **improve multiple code bases** throughout their systems

1. Optimisation Tasks
2. Time Efficiency Improvements
3. Addressing Scalability Issues
4. Future-proofing Procedures

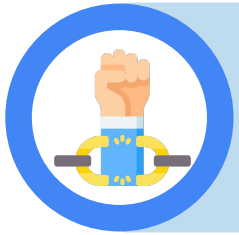
Why would developers **choose** the Sponsorship Hub?

Monetary Benefits

All jobs will pay out a minimum of $500, no matter how long it has been advertised

Direct Contact with MQ

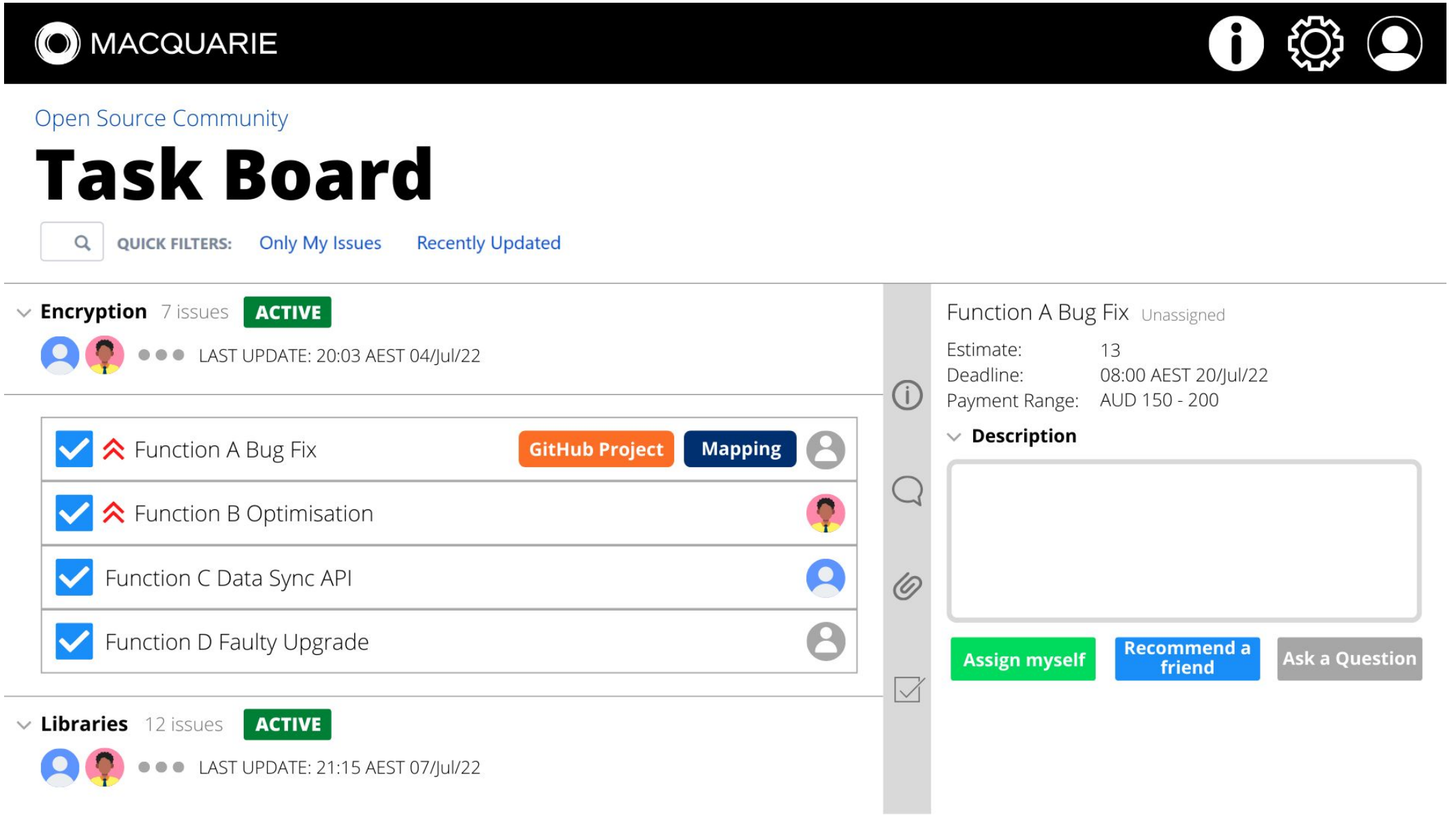
To reduce communication issues & remove barriers between developer and MQ



Maintain Freedom

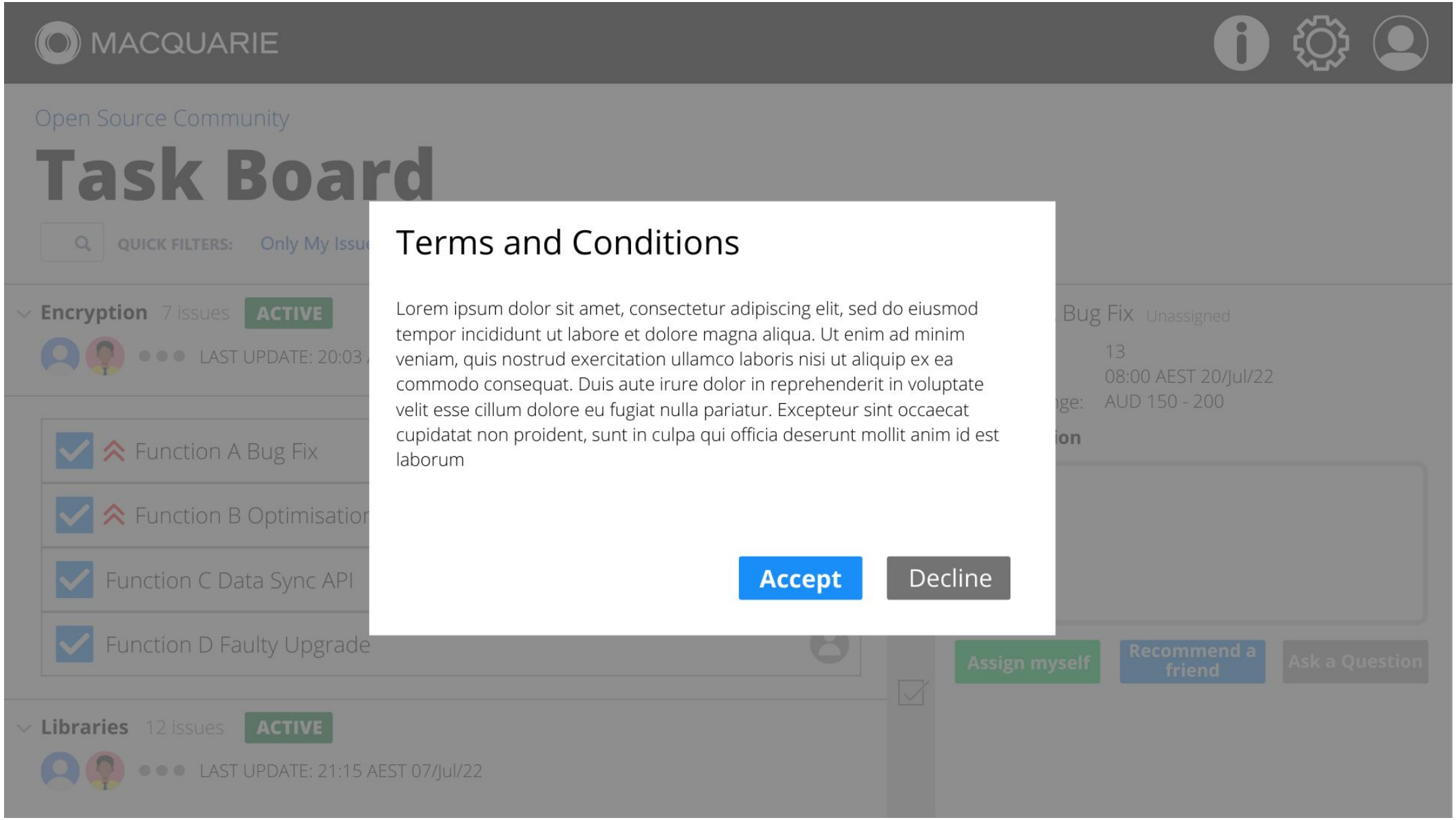
Developers still maintain all benefits of an open-source programmer if they join

Macquarie Open Source Sponsorship Hub's Interface (1)



MACQUARIE

Macquarie Open Source Sponsorship Hub's Interface (2)



MACQUARIE

Macquarie's 4-component checklist to categorise problem funding

Scale	
Quality of prescribed code Code style, length, optimisation	1 2 3 4 5
Assumed **time** needed How long does the developer need?	1 2 3 4 5
Problem **complexity** Level of understanding needed to attempt	1 2 3 4 5
Importance of task How often will Macquarie utilise this feature	1 2 3 4 5

Pricing	
17-20	$2,000
13-16	$1,500
10-13	$1,000
4-10	$500

MACQUARIE

FAQ: Macquarie is paying developers to optimise their open source software, but other people also benefit from this. Does this seem fair?

- The tasks pushed out involves the maintenance, upscaling, debugging, or optimisation of a third party library (or open source software in general) which Macquarie developers cannot handle themselves, whether it is due to time constraints, lack of understanding of code base, or any other reasons

- These third party libraries/OSS are also frequently used amongst Macquarie developers (i.e., used in multiple projects), which means that when the tasks are completed, will bring great benefits to Macquarie's existing technologies

- Because of this, ACE Advisory believes that this is the equivalent cost (or in fact, cheaper) of allocating Macquarie developers on the same tasks

- Furthermore, this will establish Macquarie as an enterprise open source supporter and contributor, which can help attract talent for Macquarie

MACQUARIE

The timely implementation of Amplification will **bulletproof** Macquarie's technologies and operations by **2025**, alongside the transition towards cloud and automation

TASKS	2022			2023			2024			2025		
	AUG	OCT	DEC	APR	AUG	DEC	APR	AUG	DEC	APR	AUG	DEC
Reliability Check Integration	Reliability Check Script Development; Database Set Up			Pipe -line Integra -tion; Script Optimisation			OSS Test Suite Development; Trial Integration		Test Suite Upgrade		Ongoing Test Upgrade & Review; Ongoing Script Optimisation and Testing	
Sysdig Secure Integration	Pipe -line Integra -tion; DEV environment roll out			PROD roll out for Advisory			Full PROD roll out for all streams			Ongoing system monitoring with support from Sysdig		
Macquarie Sponsorship Hub	Platform Development; OSS Project Preparation			OSS Developer Outreach; Beta Launch			Process and Platform Update		Full Launch		Ongoing process review & updates; Ongoing platform review & upgrade	
Staff Training	Sysdig Secure Usage Training			Ongoing Sysdig Secure Training & Review with Sysdig Support team								
				Containers Training		Security Training		Containers Training		Security Training	Containers Training	Security Training

KPIs

160+ hours saved on debugging annually

375 OSS tasks completed on MQ Hub by 2025

∅ security breach cases reported by users

LEGEND

- Primary Task
- Checkpoint
- Follow Up Task

Analysis · Strategy · **Implementation** · Impact

MACQUARIE

There will be very **low startup and operating costs** due to Sysdig's **seamless integration** with MQ's existing systems, the time saved massively reducing development costs

Key Properties

CI/CD Pipeline powered by Sysdig

Sysdig Pipeline used to flag security issues by cybersecurity team

+

DevOps time spent goes from **280 hrs/yr** to **120 hrs/yr**

↓

<$20M subscription cost to Sysdig to enact this database monitoring

Open Source Sponsorship Hub

Website development and consistent maintenance by development team

+

Each problem's pay out ranges from **$500** to **$2,000**

↓

$150K investment per year to leverage the open-source community

Immediate Impacts

≈ 70 M saved By **2025** due to Sysdig Pipelines

Estimation of **375** problems to be solved by **2025**

Expenses

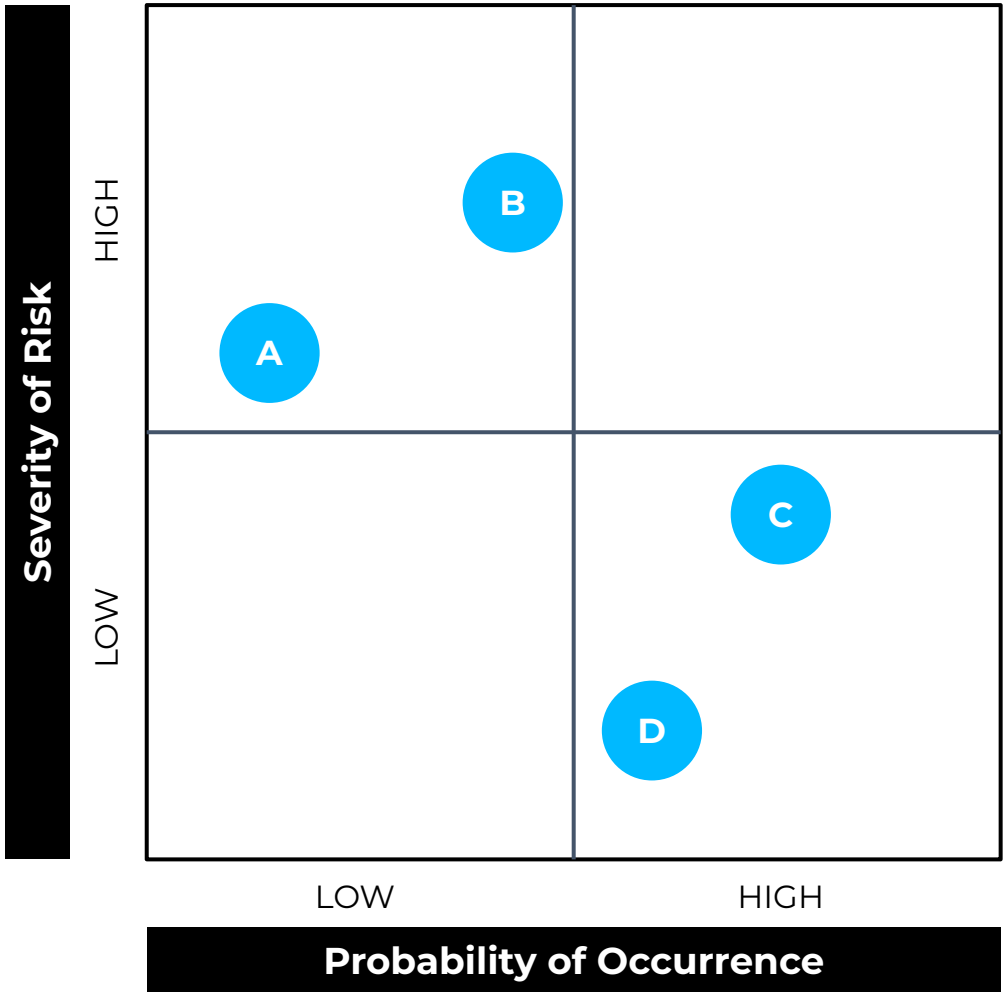
EXPENSES						
1. Sysdig Secure	Units	2022	2023	2024	2025	Source
Cloud SIEM cost	$/GB of analysed logs/month	$0.20				Real-time security detector cost taken from one of Sysdig's competitors, DataDog, https://www.datadoghq.com/pricing/?product=cloud-siem#cloud-siem
	$/GB of analysed logs/year	$2.40	$2.57	$2.75	$2.94	
Average GB of analysed logs in a single project	GB	93				https://becomingdevs.com/how-much-storage-does-a-programmer-need/#:~:text=According%20to%20the%20size%20estimation%2C%20this%20would%20be,Space%20required%20for%20web%20and%20mobile%20app%20programming.
# of MQ OSS associated projects	#/year	6240	6952	7744	8624.88	
# GB of analysed logs throughout the year	GB	580320	646536	720192	802113.84	
Assumed amount of times spent re-entering into pipeline	#	15				
Total Cloud SIEM Subscription Cost	$	$20,891,520.00	$24,904,566.72	$29,683,721.55	$35,374,462.02	
Database monitoring costs	$/database host/month	$70.00				Normalised stability and performance metrics taken from DataDog https://www.datadoghq.com/pricing/?product=database-monitoring
	$/database host/year	$840.00	$898.80	$961.72	$1,029.04	
Total Cost	$	$20,892,360.00	$24,905,465.52	$29,684,683.26	$35,375,491.05	
2. Reliability Check Script Development	Units	2022				
Average developer pay	#	4				
	$/hr	$36.00				https://www.payscale.com/research/AU/Job=Software_Developer/Salary
# of developing hours	hr/week	16				Assumption of 16 hours a week per person to develop the script
# of development weeks	week	9				3 month development lifecycle
Total Cost	$	$20,736				
3. Macquarie Open Source Sponsorship Hub	Units	2022	2023	2024	2025	Source
# of platform maintenance staff	#	6	4	4	4	2 extra staff in the first year in order to create the website
Average pay	$/hr	$36.44	$38.99	$41.72	$44.64	https://www.payscale.com/research/AU/Job=Website_Manager/Salary
# of hours in uploading problems & tasks	hr	500	1500	1248	1248	https://spdload.com/blog/average-time-to-create-a-website/
Total Cost	$	$18,220.00	$58,486.20	$52,066.75	$55,711.43	1500 development hours for beta release maintenance in 2023, 20 hours a week to maintain requests between 4 staff from 2024 onwards
Sponsorship Money						
Average payment	$	1025	1025	1025	1025	Assuming problem split of: 40% - $500, 25% - $1000, 25% - $1500, 10% - $2000
Problems per year	#	0	75	150	150	Assume 0.5 problem per day during beta, 3 problem per week from 2024 onwards on average
Total Cost	$	$0	$76,875	$153,750	$153,750	
TOTAL COST	**$**	**$20,931,316**	**$25,040,827**	**$29,890,500**	**$35,584,952**	

Cost Savings Calculation

IMPROVED TIME EFFICIENCY OF DEVELOPERS						
	Units	2022	2023	2024	2025	Source
# of projects per developer	#/3 months	5	5	5	5	https://www.quora.com/Whats-the-average-number-of-projects-a-software-engineer-is-expected-to-work-on-concurrently-at-their-job
	#/year	20	20	20	20	
# of MQ developers	#	2000	2200	2420	2662	Assuming that MQ technology team's size grow 10% per year https://www.macquarie.com/au/en/careers/our-people/where-could-a-career-in-technology-take-you.html#:~:text=Our%20Technology%20team%20comprises%20over,invest%20for%20a%20better%20future
# of MQ development projects	#/year	8000	8800	9680	10648	
% of OSS associated projects	%	78%	79%	80%	81%	
# of MQ OSS associated projects	#/year	6240	6952	7744	8624.88	
% of buggy/failed projects	%	70%	70%	70%	70%	https://teamstage.io/project-management-statistics/
# of buggy OSS projects	#/year	4368	4866.4	5420.8	6037.416	
# of developers per project bug fix	#	2	2	2	2	Assuming that there are 5 developers per project
Average developer pay	$/hr	36	36	36	36	
Time spent on bug fix (without solution)	month	7	7	7	7	Developers take 7+ months to fix 50% of flaws if undetected https://www.cpomagazine.com/cyber-security/open-source-security-flaws-exist-in-70-of-applications-80-of-libraries-are-never-updated/
	hr/month	40	40	40	40	
	hr/year	280	280	280	280	
# of hours spent on bug fix (with solution)	hr/month	10	10	10	10	Developers can take between 1 day to 1 week to fix flaws if they're aware of it Taking upper range for estimation (i.e. 1 week) https://www.cpomagazine.com/cyber-security/open-source-security-flaws-exist-in-70-of-applications-80-of-libraries-are-never-updated/
	hr/year	120	120	120	120	
Cost of bug fix without solution	$	$88,058,880	$98,106,624	$109,283,328	$121,714,307	
Cost of bug fix with solution	$	$37,739,520	$42,045,696	$46,835,712	$52,163,274	
Savings	**$**	**$50,319,360**	**$56,060,928**	**$62,447,616**	**$69,551,032**	

MACQUARIE

Amplification is not without **uncertainty**, however, they can be easily counteracted through **prepared strategies**



A	Undetected vulnerabilities & security flaws by Sysdig Secure	Post deployment security scanning can expose previously undetected threats
B	Vulnerable periods of Sysdig Secure downtime	Revert back to original security system during downtime
C	Mundane bounties are left ignored by developers	Ignored bounties will be passed onto Macquarie DevOps teams
D	Low variance of bounties and exploitation	Similar bounties to previous cases will have reduced prices