Threat Modelling in the Cloud

Roy Harrow - *BCS DevSecOps Group* - 22nd May 2025

Roy Harrow

- Applications development
- Methods and Tools
- Standards and procedures
- Change and configuration management
  - Secretary of BCS CMSG
- Design and Architecture
- Security
  - Web Security and IAM
  - Infrastructure / Designed DIY monitoring system
  - Consultancy and architecture
  - Solution design
  - Product assurance
  - Cloud Security

- Sectors
  - Financial services
  - Central and Local Government
  - Communications, Health Care, Transport, Retail + many others

- IBM Security
  - 2008-2022 (14 years)
- Sainsbury's Information Security
  - September 2022+

- Chair of BCS DevSecOps group

- Huge range of business applications
  - eCommerce: groceries online, general merchandise
  - Store systems, point of sale, warehouse, delivery and logistics
  - Contact centre, corporate services many more
- Wide range of technologies
  - On premise: mainframe, midrange and specialised technologies
  - Cloud hosted services and many SaaS applications
- Large number of engineering teams
  - Linked to product managers
  - Using modern agile practices and CI/CD pipelines integrated with various security processes and tooling

Sainsbury's Argos NEXT 360 Sainsbury's Bank habitat

https://sainsburys.jobs/roles/digital-tech-data/

Agenda

- An Introduction to Threat Modelling
- The Cloud Threat Landscape
- Some Threat Modelling Techniques
  - Including STRIDE
- Some Freely Available Threat Modelling Tools
- Demonstrations of some Cloud Threat Modelling Scenarios and Tools
- **Introducing “Shifting the Left, Left”**
- Best Practices and Resources
- Summary & Conclusions

These are my personal views and don’t represent policies and processes from my current or previous employers

What is Threat Modelling?

A process to attempt to identify security weaknesses in an application

- ***Before someone else does***

Aims to help to improve the security of IT applications

- Ideally ***before they are built***

The focus tends to be on thinking about deliberate attempts to circumvent an application's security controls - aka "threats"

- But also needs to consider accidents
- Deliberate attempts could be targeted or just "random" / opportunistic

Why do Threat Modelling?

To identify potential vulnerabilities **early**

Ideally during design stages

To be able to influence design and build before it is too late

To include input from all stakeholders

To ensure all "angles" are considered, both technical and non-technical

To drive security controls based on business priorities

By taking inputs from product owners and business representatives

To encourage a "security mindset"

To influence the selection and design of future IT services

To help cultivate security champions

Threat Modelling – Main Steps

- Understand the system or application context
  - Business purpose
  - Information being processed
  - Any business drivers for:
    - High confidentiality, integrity or availability
- Consider what could go wrong
- What can we do about it?
- And finally
  - How did we do?

Threat Modelling – Key Activities

- We need to understand the system or application
  - Business purpose and information being processed

- Scope + Context (Business + Technical)
  - A sprint or a component
  - Data Flow Diagrams are common

- Then need to consider what could go wrong

- Brainstorm possible threats or attacks
  - Application profiling questions
  - Common threat/attack models

- What can we do about it?

- Identify or design countermeasures
  - to reduce risk

- And finally….
  - How did we do?

- “Fit for purpose” given context?
  - Coverage
  - Lessons learned

OWASP Threat Modelling Method

- Decompose the Application
  - External Dependencies
  - Entry Points and Exit Points
  - Assets
  - Trust Levels
  - Data Flow Diagrams

- Identify and Rank Threats
  - Threat Categorisation e.g. using STRIDE

- Determine Countermeasures and Mitigation
  - Typically uses the OWASP Application Security Framework (ASF) or
  - STRIDE threat mitigations

https://owasp.org/www-community/Threat_Modeling

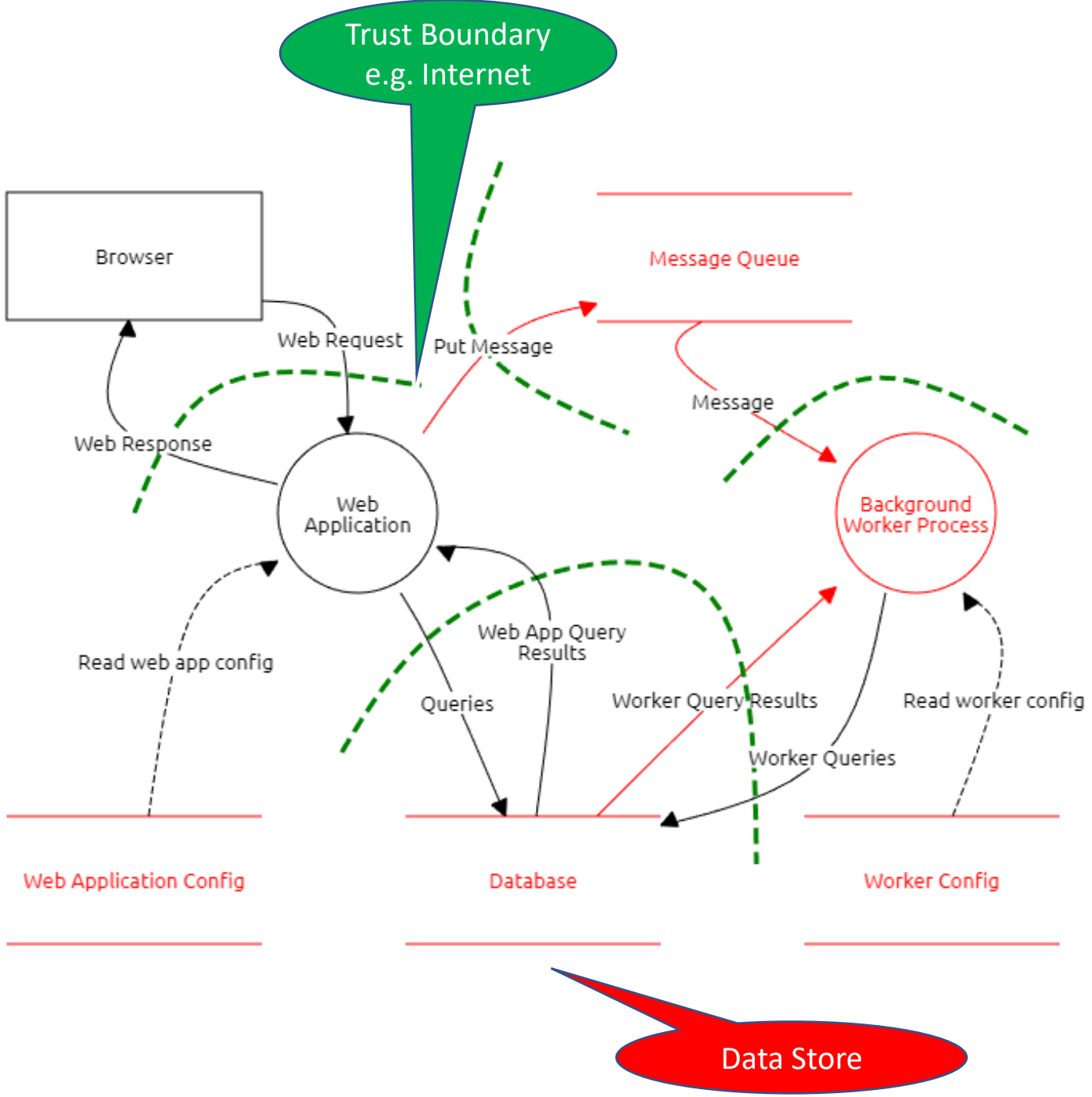
Threat Modelling Diagrams

- Useful for Scoping and Identifying Potential Targets for Attack
  - Data Flow Diagram (DFD)
  - Process Flow Diagram (PFD)
  - C4 Model - architectural diagrams
    - Context, Container, Component and Code

- Attack Tree Diagrams
  - Explains the steps of an attack
    - Bruce Schneier, 1999
    - Synopsis, 2015

Data or Process Flow Diagrams

- Helps define scope
- Aids understanding of data flows
- Provides structure for assessing risks
- Data Flow Diagrams (DFD) are the most common, for example



Threat Modelling Techniques

Threat Identification

- **Q. What might cause us to breach…. ?**
- CIA - Confidentiality, Integrity and Availability
- Compliance framework

Threat Classification

- **Q. Could we be vulnerable to certain types of attack?**
  - STRIDE
    - Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege
  - OWASP lists of top 10 types of common security vulnerabilities
    - Browser-based web applications
    - Mobile applications
    - API services

CIA approach to Threat Modelling

Confidentiality

- Will we be storing or handling any sensitive information?
- How will we be protecting it?

Integrity

- What are the consequences of an accidental or deliberate data corruption of unauthorised change?
- Why might someone want to change some data?
- What controls exist to prevent or detect unauthorised changes?

Availability

- How long could the business operate without the system?
- Have we planned any controls to help ensure availability?

STRIDE

	Threat	Property Violated	Threat Definition
S	Spoofing Identity	Authenticity	Pretending to be something or someone other than yourself
T	Tampering with data	Integrity	Modifying data at rest, in transit or in memory
R	Repudiation	Non-repudiation	Denying that you did something
I	Information disclosure	Confidentiality	Giving sensitive information to someone not authorised
D	Denial of service	Availability	Exhausting computing resources needed to support the service
E	Elevation of privilege	Authorisation	Allowing someone to do something they are not authorised to perform

https://www.microsoft.com/en-us/security/blog/2007/09/11/stride-chart/

Common Mitigations by type of Threat

Type of Threat	Mitigation Strategy / Security Controls / Countermeasures
Spoofing	Strong authentication Digital signatures Protection of secrets
Tampering	Access controls Check-sums, hash-totals and signatures on data items
Repudiation	Strong authentication Audit logs – with verifiable time-stamps Digital signatures
Information Disclosure	Access controls Encryption
Denial of Service	Quotas / throttling of transaction volumes Authentication and authorisation
Elevation of Privilege	Access controls supporting least privilege Hardened system configuration

OWASP Top 10 Proactive Security Controls

A useful source of inspiration when designing countermeasures

- C1: Define Security Requirements
- C2: Leverage Security Frameworks and Libraries
- C3: Secure Database Access
- C4: Encode and Escape Data
- C5: Validate All Inputs
- C6: Implement Digital Identity
- C7: Enforce Access Controls
- C8: Protect Data Everywhere
- C9: Implement Security Logging and Monitoring
- C10: Handle All Errors and Exceptions

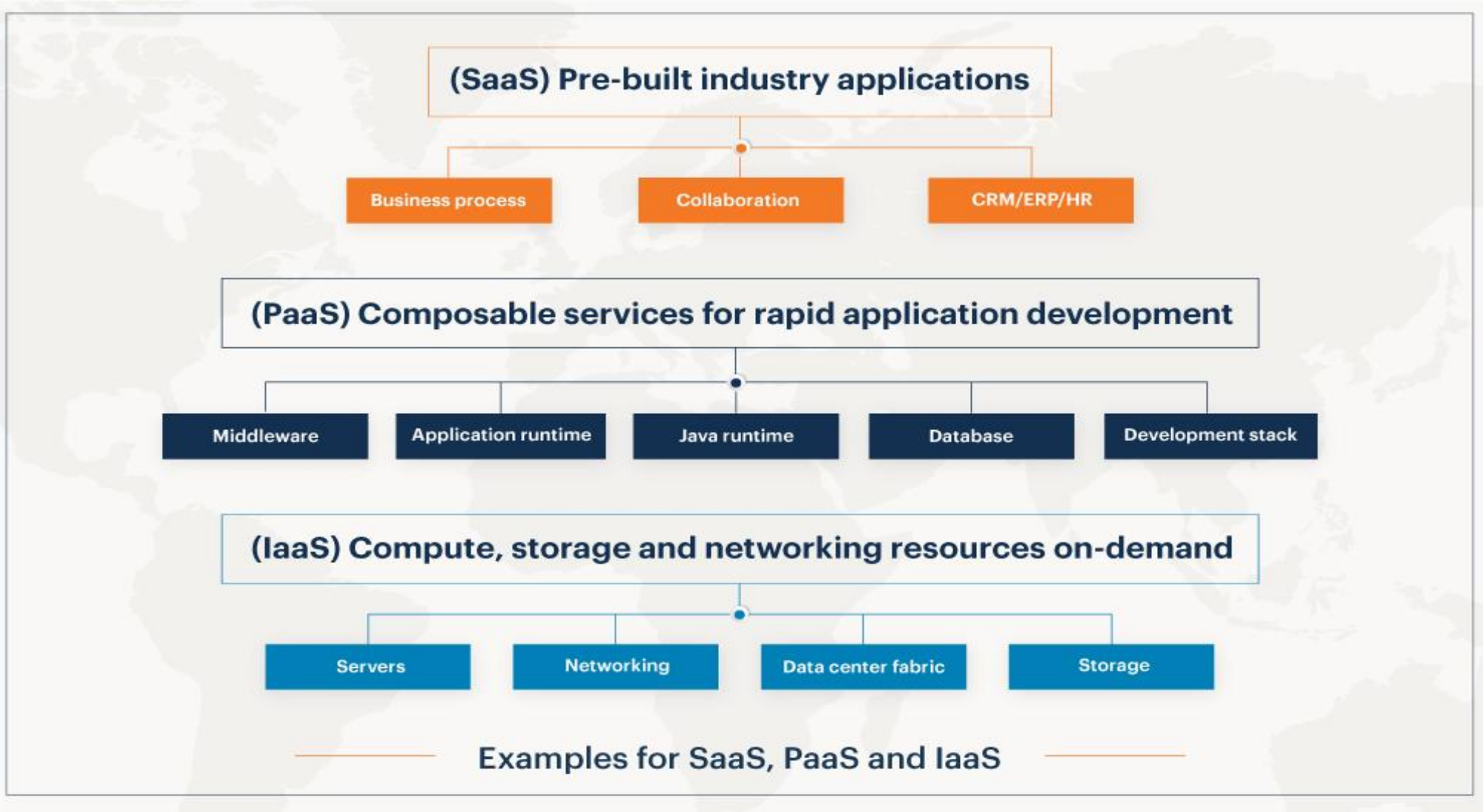
Examples of OWASP Cheat Sheets

- **Topical advice for Developers**
  - Authentication
  - Authorisation
  - Cryptographic Storage
    - Encryption of data at rest
  - Database Security
  - Docker and Kubernetes Security
  - Input Validation
  - Secrets Management

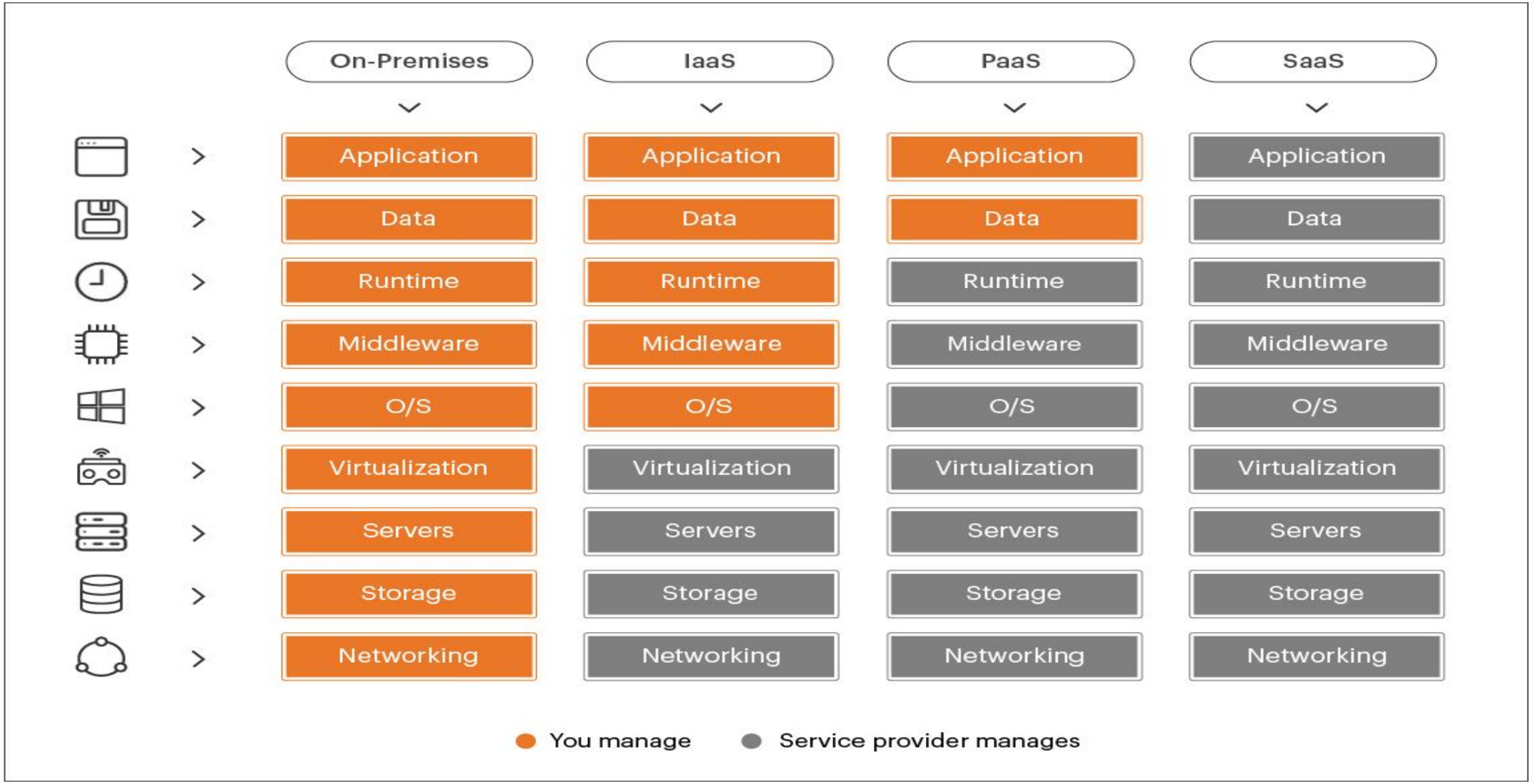
Further detailed sources of inspiration for countermeasures

- **Advice on defending against common vulnerabilities**
  - Clickjacking Defence
  - Cross Site Scripting Prevention
  - Denial of Service Protection

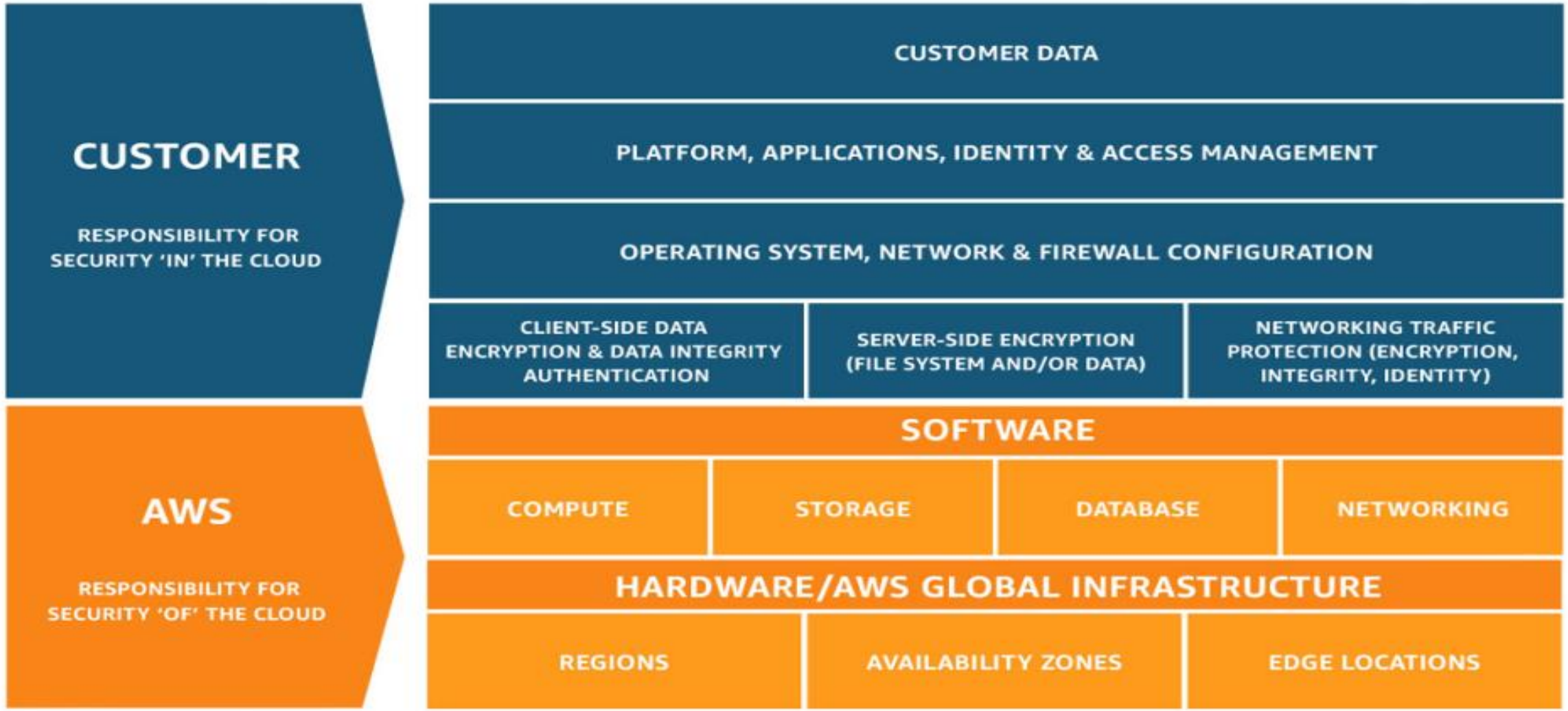
Cloud Deployment Models



Cloud Shared Responsibility Model



AWS Cloud Shared Responsibility Model



Common Threats to Cloud Applications

OWASP lists of top 10 types of common security vulnerabilities	• Browser-based web applications • Mobile applications • API services	Generic
Cloud Security Alliance	• Top Threats to Cloud Computing 2024	Cloud
OWASP Cloud-Native Application Security Top 10	• Draft recommendations only at this point	Cloud
Lists of Common Cloud Misconfigurations	• Cloud Defense.AI list of 13 • List from UpGuard	Cloud

Cloud Security Alliance (CSA) Top Threats to Cloud Computing 2024

The top Four

https://cloudsecurityalliance.org/artifacts/top-threats-to-cloud-computing-2024

1. Misconfiguration and inadequate change control
   - Cloud computing has compounded the challenges of configuration management
   - *Hence the rise of tools for Cloud Security Posture Management (CSPM) and Secrets Management*
2. Identity and Access Management (IAM)
   - Accidental data disclosure
   - Excessive Permissions
3. Insecure interfaces and APIs
   - Inadequate authentication and authorisation
   - Insufficient validation of inputs
4. Inadequate selection/implementation of cloud security strategy
   - Inappropriate use of cloud services
   - Incorrect assumptions about shared responsibility model

Cloud Security Alliance (CSA) Top Threats to Cloud Computing 2024

The bottom seven

https://cloudsecurityalliance.org/artifacts/top-threats-to-cloud-computing-2024

5. Insecure Third-Party Resources
   - Incorrect assumptions about shared responsibility model
6. Insecure Software Development
   - Lack of a robust secure development life-cycle
7. Accidental Data Disclosure
   - Open access to cloud resources such as S3 buckets due to misconfiguration
8. System Vulnerabilities
   - Making cloud hosted applications more susceptible to attack
9. Limited Cloud Visibility/Observability
   - Insufficient monitoring or alerts
10. Unauthenticated Resource Sharing
    - Access to data possible without authentication
11. Advanced Persistent Threats
    - Ransomware as a service

13 Most Common Misconfigurations on The Cloud and Their Solutions

CloudDefense.AI

https://www.clouddefense.ai/common-misconfigurations-on-the-cloud/

1. Excessive Permissions
2. Unrestricted Open Network Ports
3. Exposed Storage Buckets
4. Absence of Logging and Monitoring
5. Open ICMP
6. Keeping Default Credentials
7. Keeping Development Configuration in Production
8. Extensive Access to HTTPS and Non-HTTP Ports
9. Neglecting Safe Configuration For Third-Party Components
10. Poorly Configured Automated Backup
11. Lack of Network Segmentation
12. Weak Password Policies
13. Insecure API Configurations

Common Cloud Misconfigurations and How to Avoid Them

Alex Sukianto, Jan 2025, UpGuard.com

https://www.upguard.com/blog/cloud-misconfiguration

1. Unrestricted Inbound Ports
2. Unrestricted Outbound Ports
3. "Secrets" Management
4. Disabled Monitoring and Logging
5. ICMP Left Open (ping protocol)
6. Insecure Automated Backups
7. Storage Access – errors in access policies
8. Lack of Validation of Cloud configuration
9. Unlimited Access to Non-HTTPS/HTTP Ports
10. Overly Permissive Access to Virtual Machines, Containers, and Hosts
11. Enabling Too Many Cloud Access Permissions
12. Subdomain Hijacking (AKA Dangling DNS)
13. Misconfigurations Specific to Your Cloud Provider(s)

Examples of Cloud Security Failings

DarkBeam Data Leak 2023 - https://www.cshub.com/data/news/darkbeam-data-leak

- More than 3.8 billion records have been exposed after digital protection firm DarkBeam left an interface containing the exposed records unprotected.
- DarkBeam had been collecting the data to alert its customers in the case of a data breach, meaning the data exposed was data already leaked in prior cyber attacks. Of the data leaked, there were 16 collections named ‘email 0-9' and ‘email A-F' which represented 239,635,000 pairs of login credentials.
- The data leak was caused by leaving a Elasticsearch and Kibana data visualization interface unportected, allowing access to the confidential data held within it.

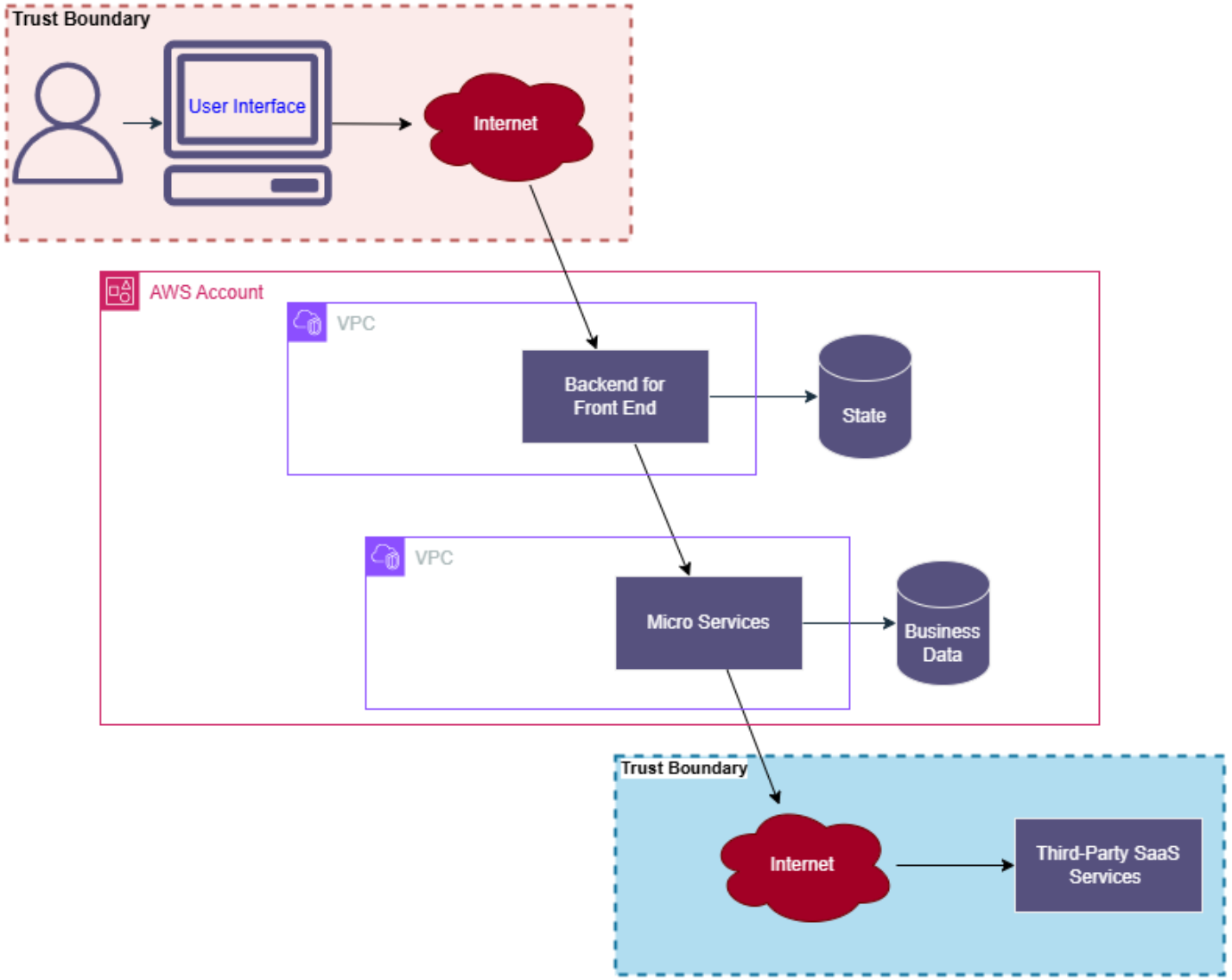
Toyota Data Leaks

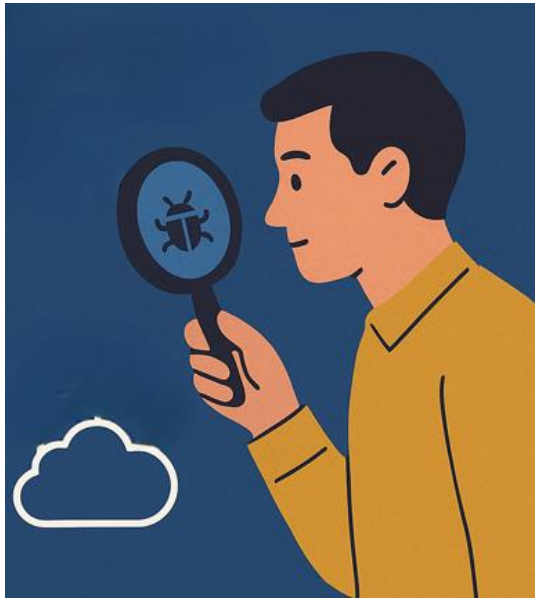
- 2022 Toyota admitted that it had stored the data of over two million drivers – including vehicle location data – on a publicly available cloud database for over a decade owing to human error that went undetected.
- 2023 two misconfigured cloud services were found leaking 260,000 car owners' personal information over a seven-year period. Customers' information such as names, phone numbers, email addresses, and vehicle registration numbers may have been externally accessible from October 2016.

Snowflake

- 2024 - June 2024, Mandiant researchers warned that a threat actor had stolen a significant volume of customer data from multi-cloud data warehousing platform Snowflake using stolen customer credentials.
- The data was advertised for sale on cybercrime forums as well as the threat actor using the data in attempts to extort many of the victims.

Example Tiered Cloud Hosted Web-Application

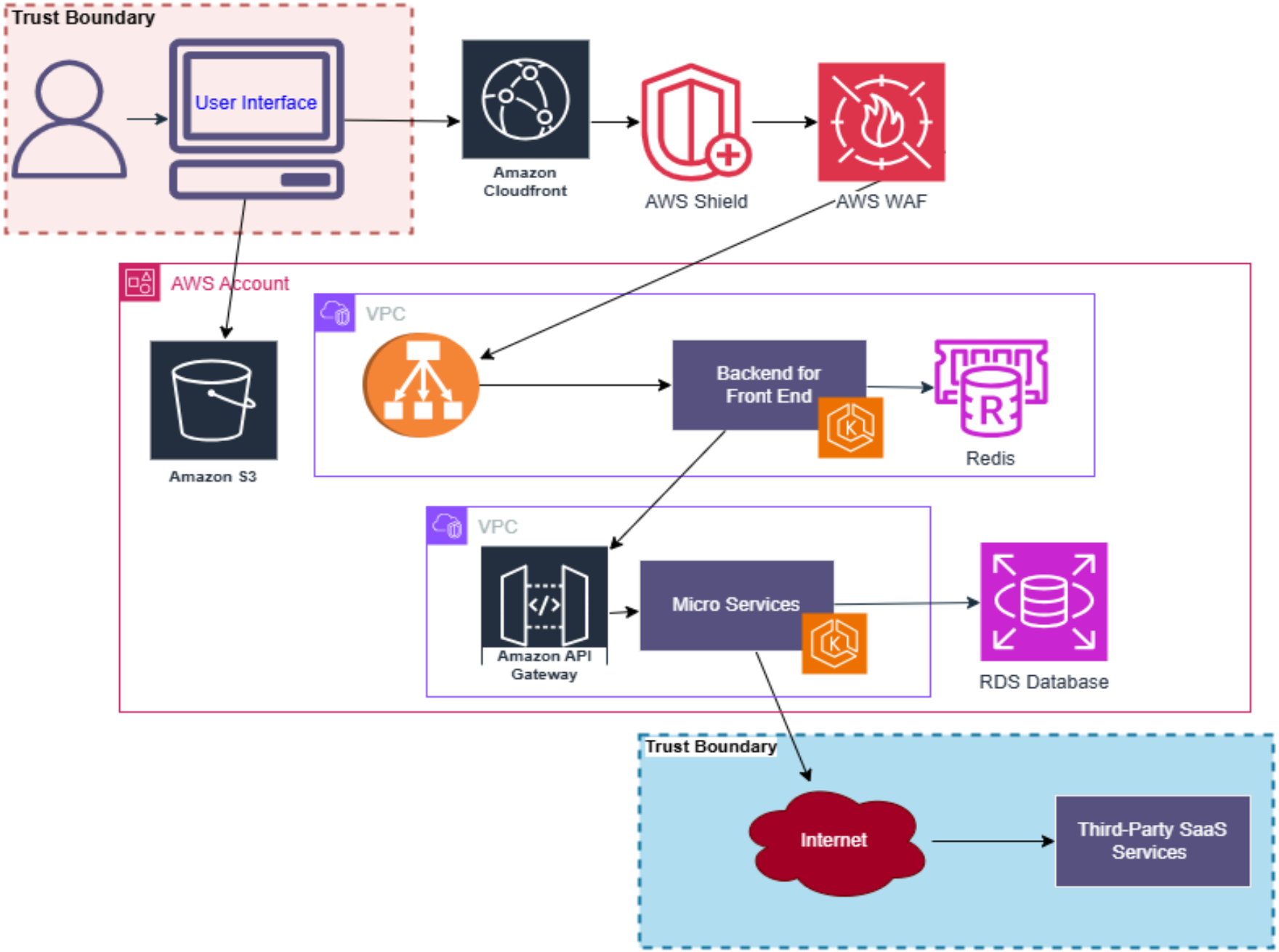




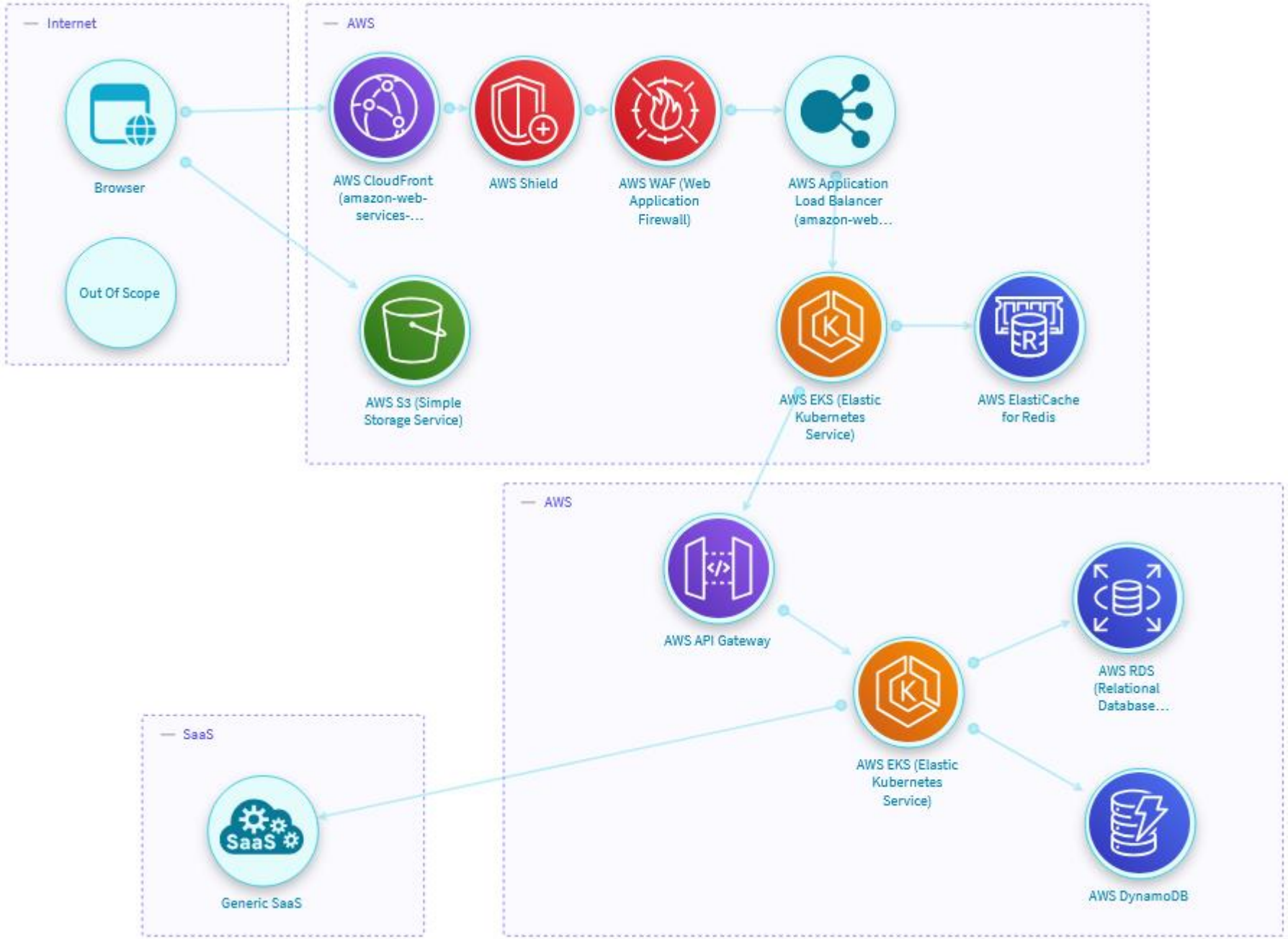
Tiered Cloud Hosted Web-Application Including Example AWS Cloud Services

- Browser User Interface
  - JavaScript + static content from S3 bucket
- Web and Application Server Components
  - Backend for Front-end (BFF)
    - Accessed via Amazon Cloudfront, AWS Shield, AWS WAF and Application Load Balancer (ALB)
    - Hosted as AWS Lambda, ECS or EKS
    - State information in AWS Redis
  - Microservices
    - Accessed via AWS API Gateway
    - Hosted as AWS ECS or EKS
  - Business Data Store e.g. DynamoDB, Amazon RDS database
- Third-Party SaaS applications
  - e.g. Payment Processing

Example AWS Services



IriusRisk Diagram of Tiered Web Application Example



Threat Modelling by Cloud Layer - IaaS

- Infrastructure as a Service (IaaS)
  - Examples: AWS VPCs, compute and network resources.
  - Uses Terraform or Cloud specific Infrastructure as Code (IAC)
  - Common threats:
    - overly permissive access
    - unencrypted storage
    - open ports
- Mitigation Tools:
  - Checkov (IaC scanning): https://www.checkov.io/
  - tfsec (Terraform security scanner): https://aquasecurity.github.io/tfsec/
  - Cloud Security Posture Management Tools – run-time environment

Threat Modelling by Cloud Layer - PaaS

- Platform as a Service (IaaS)
  - Example services:
    - Kubernetes e.g. Amazon EKS
    - Service mesh e.g. ISTIO
    - Observability stack
  - Common threats:
    - Insecure control planes
    - Excessive privileges
    - Compromised CI/CD.
  - Mitigation Tools:
    - Kubescape: Kubernetes posture scanning.
    - Trivy: container and IaC vulnerability scanner
    - Cloud Security Posture Management Tools – run-time environment

Threat Modelling the Application Layer - UI

- User Interface Layer or Frontend (Browser/SPA/Static Web App)
  - Typically deployed via S3 + AWS CloudFront. Communicates with BFF or APIs via HTTPS.
- Common Threats
  - Cross-site scripting (XSS) from unsafe DOM manipulation or unescaped data.
  - Cross-site request forgery (CSRF) in form submissions.
  - Clickjacking using iFrames.
  - Data leakage via browser storage (e.g., JWTs in localStorage).
  - Open redirect or unvalidated input from URL parameters.
- Mitigations (Design + Cloud Services)
  - Use CSP (Content Security Policy) headers via CloudFront/ALB.
  - Enable WAF XSS and SQLi rule sets at CloudFront or ALB.
  - Apply Subresource Integrity (SRI) on third-party scripts.
  - Set SameSite and Secure attributes on cookies.
  - Obfuscate user data sent to frontends.

Threat Modelling the Application Layer - BFF

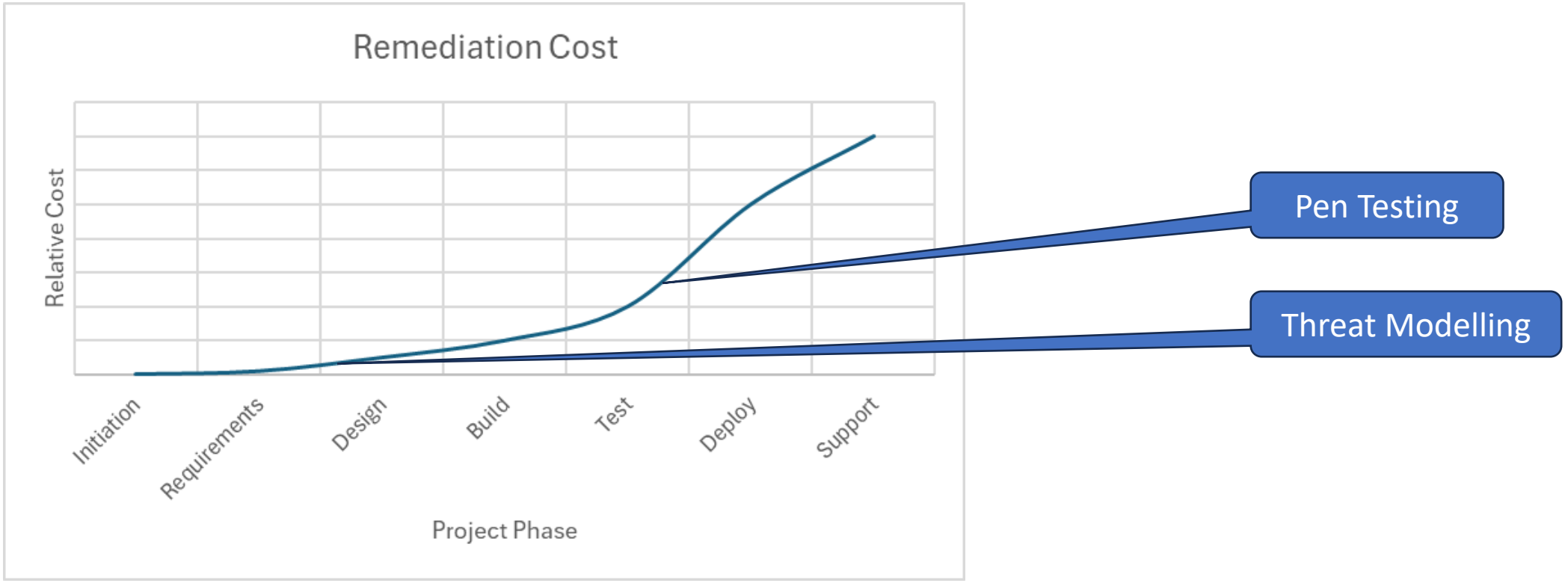
- Backend-for-Frontend (BFF)
  - Often deployed via Lambda behind API Gateway or EKS services. Serves tailored responses to frontend based on user context.
- Common Threats
  - Authentication bypass due to poor session handling.
  - Improper authorization (e.g., failing to check roles/scopes).
  - Parameter pollution or input tampering.
  - Leaky APIs that expose internal system details (overly verbose error messages).
  - Abuse of client-controlled input to manipulate logic.
- Mitigations (Design and Cloud Services)
  - Use Amazon Cognito JWT validation middleware or Lambda authorizers.
  - Normalize and validate all input.
  - Return minimal error information (use error codes, not stack traces).
  - Rate-limit via API Gateway usage plans or WAF throttling.
  - Log securely to CloudWatch, masking PII.

Threat Modelling the Application Layer - Microservices

- Includes APIs, authentication, messaging queues, and event handlers used across microservices and environments.
- Example Cloud Services:
  - API Gateway endpoints
  - Amazon Cognito for Authentication and Authorisation
  - SNS/SQS for async messaging
  - Kubernetes such as Amazon EKS, Lambda functions
- Common Threats
  - Broken authentication/authorization across services.
  - Replay attacks on signed or tokened requests.
  - Insecure event/message ingestion, leading to injection or event flooding.
  - Lack of segregation, enabling lateral movement between tenants.
  - Secrets exposure in logs, environment variables or event payloads.
- Mitigations
  - Use Cognito with for API-level access control.
  - Validate incoming events using message signatures or structured schemas (e.g., JSON Schema).
  - Enforce resource-level IAM policies.
  - Use Secrets Manager or Parameter Store for environment config.
  - Enable encryption in transit and at rest for all data and messages.

Traditional Shift Left Model

- Aims to save time and costs by identifying and fixing security issues earlier in a project and not leaving this until the pen-test at the end



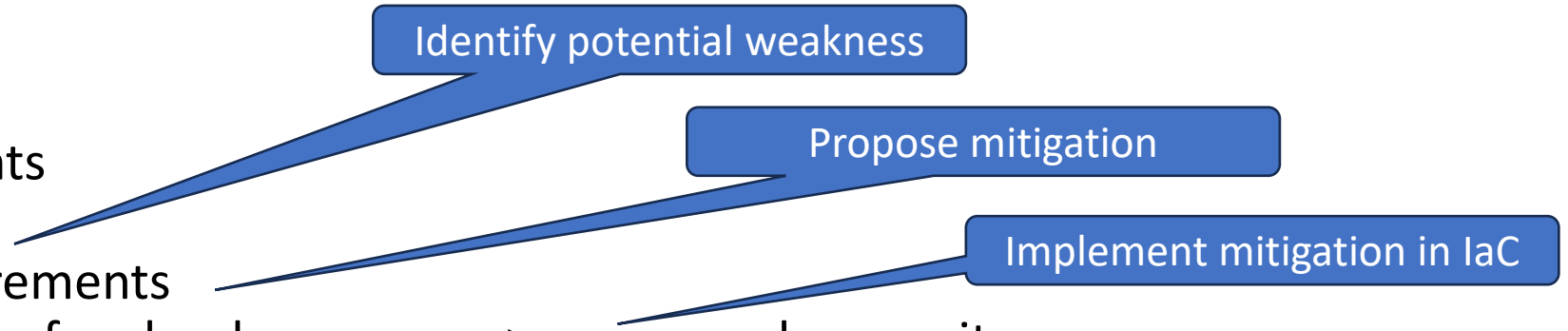
Use of IaC for Cloud Projects

- Infrastructure as Code (IaC) is very commonly used to automate the deployment of cloud resources
- Steps
  1. Project requirements
  2. High-level design
  3. Define cloud requirements
  4. Create IaC definition for cloud resources -> source code repository
  5. Automated deployment of cloud resources
  6. Build and deploy application code into cloud

Threat Modelling Opportunity with IaC

Steps

1. Project requirements
2. High-level design
3. Define cloud requirements
4. Create IaC definition for cloud resources -> source code repository
5. Automated deployment of cloud resources
6. Build and deploy application code into cloud

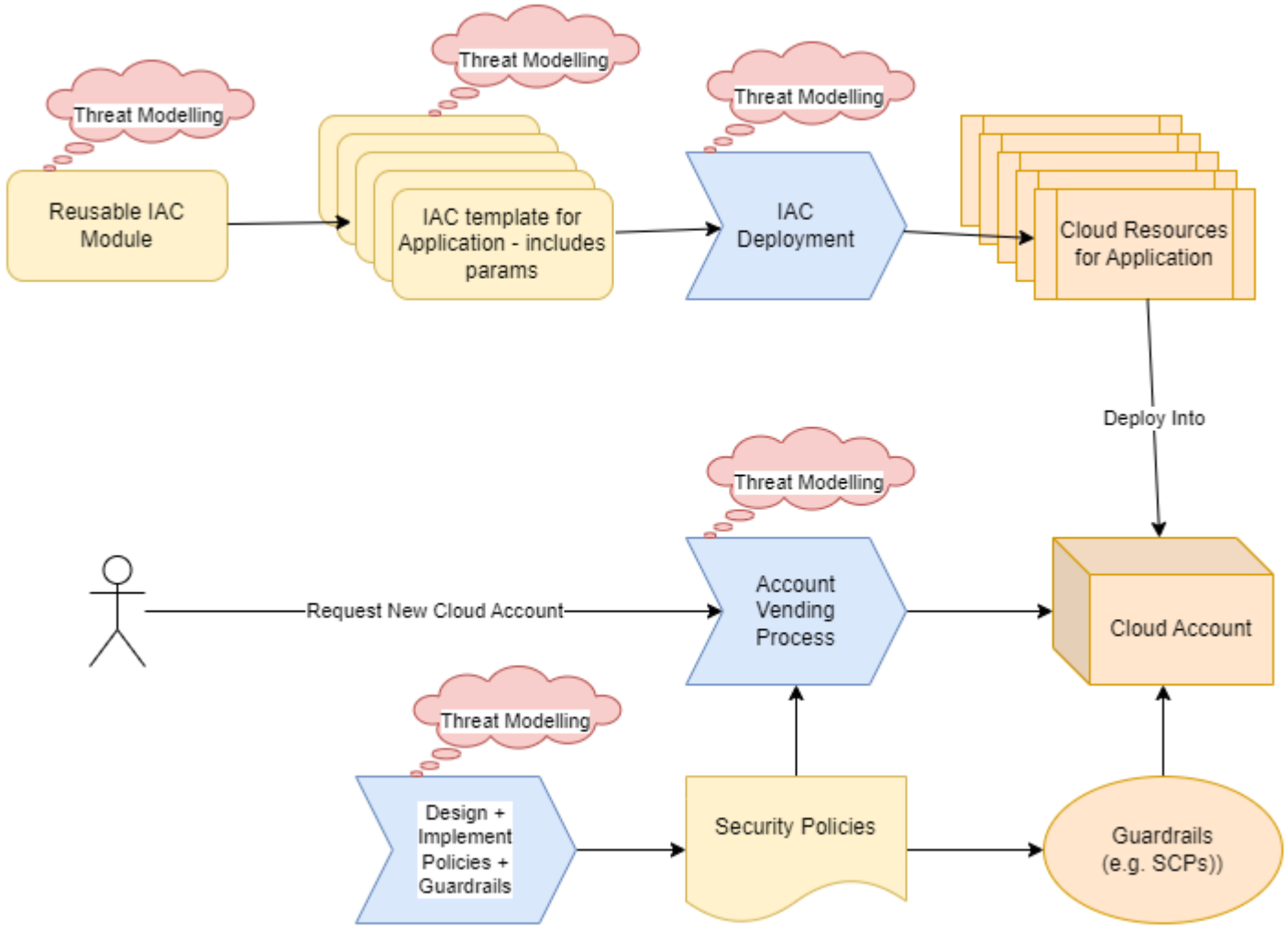


For example:

- Public S3 buckets are identified as a potential risk
- Review and amend IAC to ensure S3 buckets will be private
  - This may use tagging of cloud resources
- In addition, IaC can be validated for weaknesses by scanning source code (but this is not really “threat modelling”)

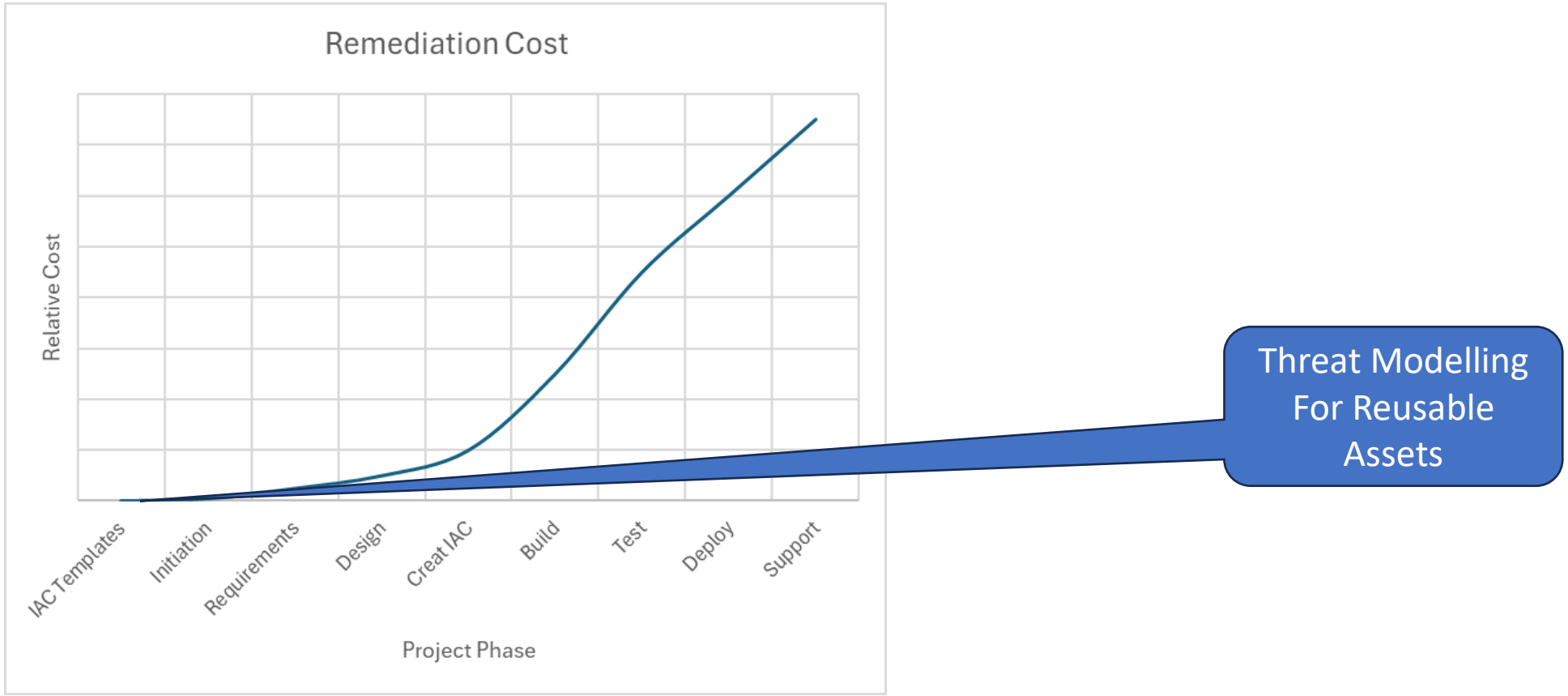
Shifting the Left, Left - in the Cloud

- Using reuseable IaC modules
- Using security guardrails
- Both offer opportunities for threat modelling



Shifting the Left, Left – Multiplication

- The benefits of threat modelling on reusable assets are multiplied across many projects



Features of Threat Modelling Tools

Systems modelling

- Typically as a flow diagram

Threat intelligence

- To inform and prompt for potential threats
- To suggest potential mitigations

Dashboard of vulnerabilities identified

- Showing severity

Dashboard of mitigations defined

- Mapping mitigations to vulnerabilities and threats
- Links to guidance for developers

Rules engine

- To add value by interpreting policies when applied to the system model

Supports Collaboration

Integration with existing processes and tools

- Issue tracking tools such as Jira
- Diagramming tools

Reporting and Exporting of Information

* e.g. diagrams

* Lists of proposed security controls

Some Threat Modelling Tools

Tool	Licensing
Microsoft Threat Modeling Tool	Free
OWASP Threat Dragon	Free
IriusRisk	Commercial + free community tier
AWS Threat Composer	Open Source
ThreatCanvas by SecureFlag	Commercial with a free “lite” version
ThreatModeler	Commercial
SD Elements by Security Compass	Commercial
Elevation of Privilege (EoP) Security Cards by Microsoft	Free
OWASP Cornucopia Card Game	Free

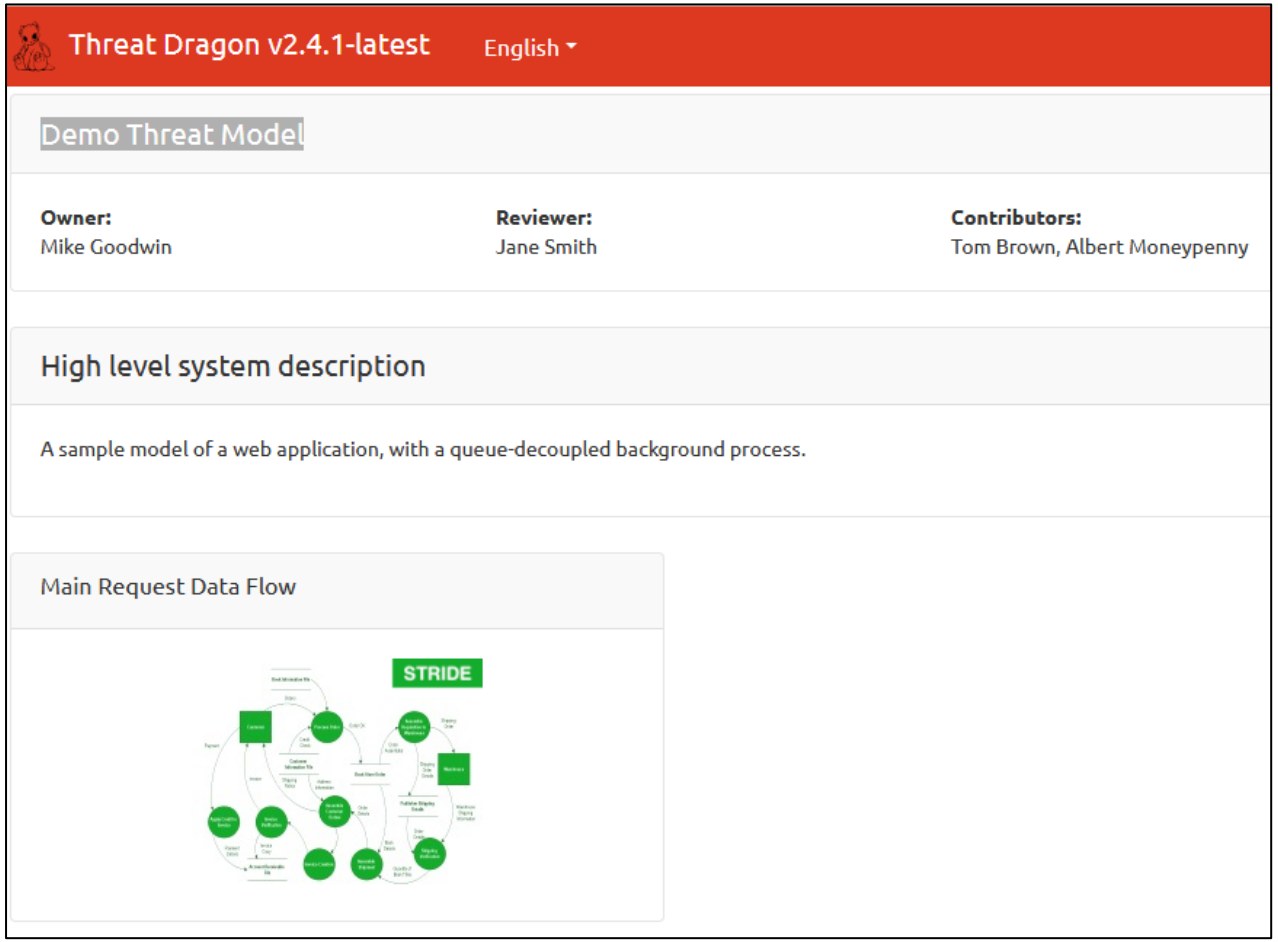
OWASP Threat Dragon

- A modelling tool used to create threat model diagrams as part of a secure development lifecycle.
- Follows the values and principles of the threat modelling manifesto.
- Can be used to record possible threats and decide on their mitigations, as well as giving a visual indication of the threat model components and threat surfaces.
- Supports STRIDE / LINDDUN / CIA / DIE / PLOT4ai
- Provides modelling diagrams and implements a rules engine to auto-generate threats and their mitigations.
- Runs either as a web application or as a desktop application.
- Online Demo: https://www.threatdragon.com/#/

OWASP Threat Dragon Demo

Using the online demo: https://www.threatdragon.com/#/

1. Select the “Demo Threat Model”



Threat Dragon v2.4.1-latest English

Entities

Components

Process

Store

Actor

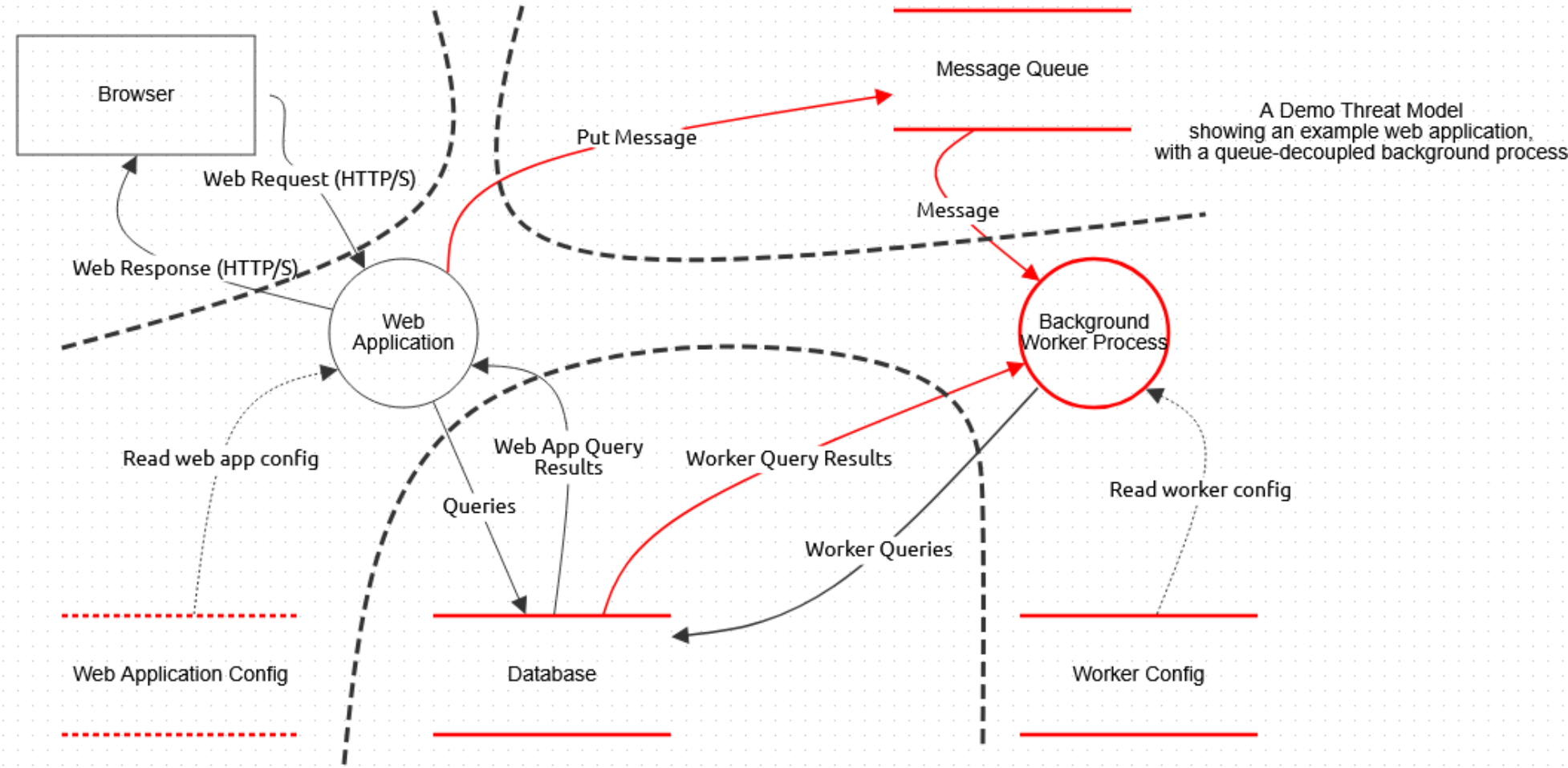
Data Flow

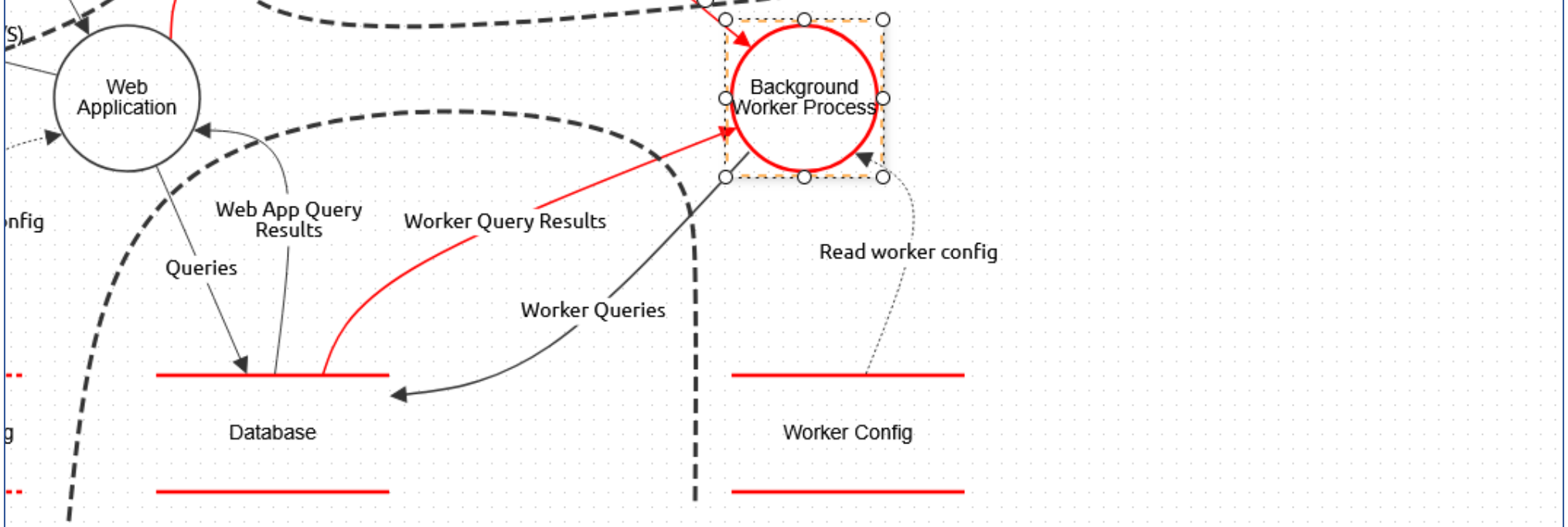
Boundaries

Trust Boundary

Metadata

Main Request Data Flow





Threats

Poison messages 1
Denial of service
STRIDE

Poison messages 2
Denial of service
STRIDE

Title

Poison messages 1

Type

Denial of service

Status

N/A | Open | Mitigated

Score

Priority

TBD | Low | Medium | High | Critical

Description

An attacker could generate a malicious message that the Background Worker cannot process.

Mitigations

Implement a poison message queue where messages are placed after a fixed number of retries.

Title

Poison messages 2

Type

Denial of service

Status

N/A | Open | Mitigated

Score

Priority

TBD | Low | Medium | High | Critical

Description

An attacker could generate a malicious message that the Background Worker cannot process.

Mitigations

Validate the content of all messages, before processing. Reject any message that have invalid content and log the rejection. Do not log the malicious content - instead log a description of the error.

OWASP Threat Dragon Demo - Report

Edit | Report | Close Model

Executive Summary

High level system description

A sample model of a web application, with a queue-decoupled background process.

Summary

Name	Value
Total Threats	14
Total Mitigated	4
Not Mitigated	10
Open / Critical Priority	0
Open / High Priority	4
Open / Medium Priority	4
Open / Low Priority	2
Open / TBD Priority	0
Open / Unknown Priority	0

OWASP Threat Dragon Demo - Report

Worker Config (Store)

Description:

Number	Title	Type	Priority	Status	Score	Description	Mitigations
	Accessing DB credentials	Information disclosure	High	Open		The Background Worker configuration stores the credentials used by the worker to access the DB. An attacker could compromise the Background Worker and get access to the DB credentials.	Encrypt the DB credentials in the configuration file. Expire and replace the DB credentials regularly.

Database (Store)

Description:

Number	Title	Type	Priority	Status	Score	Description	Mitigations
	Unauthorised access	Information disclosure	High	Mitigated		An attacker could make an query call on the DB,	Require all queries to be authenticated.
	Credential theft	Information disclosure	Medium	Open		An attacker could obtain the DB credentials and use them to make unauthorised queries.	Use a firewall to restrict access to the DB to only the Background Worker IP address.

OWASP Threat Dragon Demo - Report

Web Application Config (Store) - *Out of Scope*

Reason for out of scope:

Description:

Number	Title	Type	Priority	Status	Score	Description	Mitigations
	Credentials should be encrypted	Information disclosure	High	Open		The Web Application Config stores credentials used by the Web App to access the message queue. These could be stolen by an attacker and used to read confidential data or place poison message on the queue.	The Message Queue credentials should be encrypted.

OWASP Threat Dragon Demo - Report

Message Queue (Store)

Description:

Number	Title	Type	Priority	Status	Score	Description	Mitigations
	Message secrecy	Information disclosure	Low	Open		The data flow between the Web Application and the Background Worker is not point-to-point and therefore end-to-end secrecy cannot be provided at the transport layer. Messages could be read by an attacker at rest in the Message Queue.	Use message level encryption for high sensitivity data (e.g. security tokens) in messages.
	Message tampering	Tampering	Medium	Open		Messages on the queue could be tampered with, causing incorrect processing by the Background Worker.	Sign all queue messages at the Web Server. Validate the message signature at the Background Worker and reject any message with a missing or invalid signature. Log any failed messages.
	Fake messages could be placed on the queue	Spoofing	High	Mitigated		An attacker could put a fake message on queue, causing the Background Worker to do incorrect processing.	Restrict access to the queue to the IP addresses of the Web Server and Background Worker. Implement authentication on the queue endpoint.

OWASP Threat Dragon Demo - Report

Background Worker Process (Process)

Description:

Number	Title	Type	Priority	Status	Score	Description	Mitigations
	Poison messages 1	Denial of service	Medium	Open		An attacker could generate a malicious message that the Background Worker cannot process.	Implement a poison message queue where messages are placed after a fixed number of retries.
	Poison messages 2	Denial of service	Medium	Open		An attacker could generate a malicious message that the Background Worker cannot process.	Validate the content of all messages, before processing. Reject any message that have invalid content and log the rejection. Do not log the malicious content - instead log a description of the error.

OWASP Threat Dragon Demo - Report

Put Message (Data Flow)

Description:

Number	Title	Type	Priority	Status	Score	Description	Mitigations
	Data flow should use HTTP/S	Information disclosure	High	Open		These requests are made over the public internet and could be intercepted by an attacker.	The requests should require HTTP/S. This will provide confidentiality and integrity. HTTP should not be supported.

Message (Data Flow)

Description:

Number	Title	Type	Priority	Status	Score	Description	Mitigations
	Data flow should use HTTP/S	Information disclosure	High	Open		These requests are made over the public internet and could be intercepted by an attacker.	The requests should require HTTP/S. This will provide confidentiality and integrity. HTTP should not be supported.

OWASP Threat Dragon Demo - Report

Worker Query Results (Data Flow)

Description:

Number	Title	Type	Priority	Status	Score	Description	Mitigations
	Man in the middle attack	Information disclosure	Low	Open		An attacker could intercept the DB queries in transit and obtain sensitive information, such as DB credentials, query parameters or query results (is unlikely since the data flow is over a private network).	Enforce an encrypted connection at the DB server

Web Response (Data Flow)

Description:

Number	Title	Type	Priority	Status	Score	Description	Mitigations
	Data flow should use HTTP/S	Information disclosure	High	Mitigated		These responses are over the public internet and could be intercepted by an attacker.	The requests should require HTTP/S. This will provide confidentiality and integrity. HTTP should not be supported.

OWASP Threat Dragon Demo - Report

Web Request (Data Flow)

Description:

Number	Title	Type	Priority	Status	Score	Description	Mitigations
	Data flow should use HTTP/S	Information disclosure	High	Mitigated		These requests are made over the public internet and could be intercepted by an attacker.	The requests should require HTTP/S. This will provide confidentiality and integrity. HTTP should not be supported.

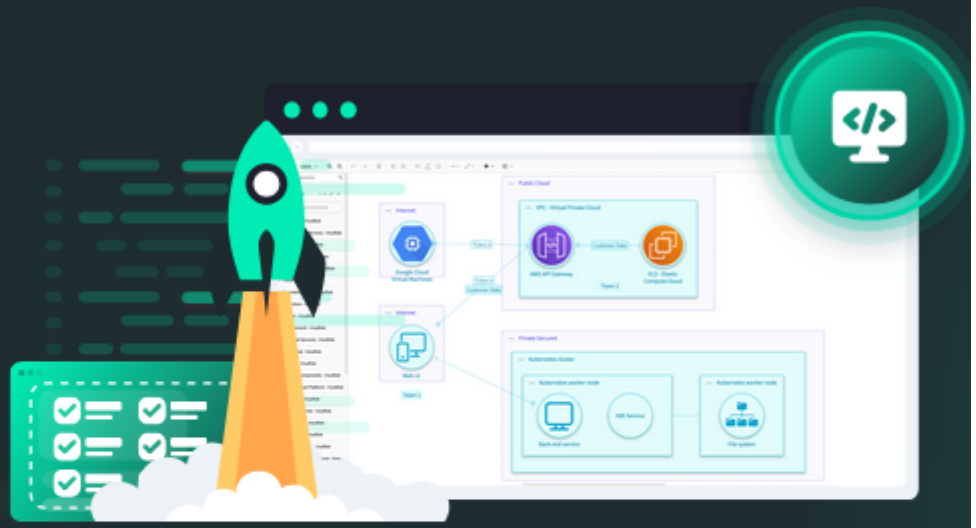
IriusRisk Community Edition

https://www.iriusrisk.com/community

IriusRisk Community: Free Threat Modeling

Get your free lifetime subscription to IriusRisk Community Edition - zero commitment access to Threat Modeling tools, libraries, and our AI assistant.

Now including THREE free threat models (easy for you to say).



Get Started for Free

Did you know, IriusRisk Community Edition is funded by the European Union?

This has meant additional features, innovative AI capability, improved UX, and more, has been possible for product development. In addition, these features have helped contribute to our 15,000 Freemium Users. We are thrilled that the funding has enabled us to get proactive security and secure by design practices, into the hands of more and more people - regardless of their security budget or previous threat modeling experience.

Co-funded by the European Union

ECCC
EUROPEAN CYBERSECURITY COMPETENCE CENTRE

Community Version Benefits

It's not just our AI Assistant you get access to in Community. Quickly and easily architect an application using our Draw.io integration and understand potential security threats and countermeasures in one simple, easy-to-use interface. You can also send a link to your Project to a friend or peer, to invite them to collaborate with you. The best part? There are no strings attached. It is truly free-forever.

Free lifetime subscription

3 threat models

Export Threats & Countermeasures as XLS

Export threat models as XML

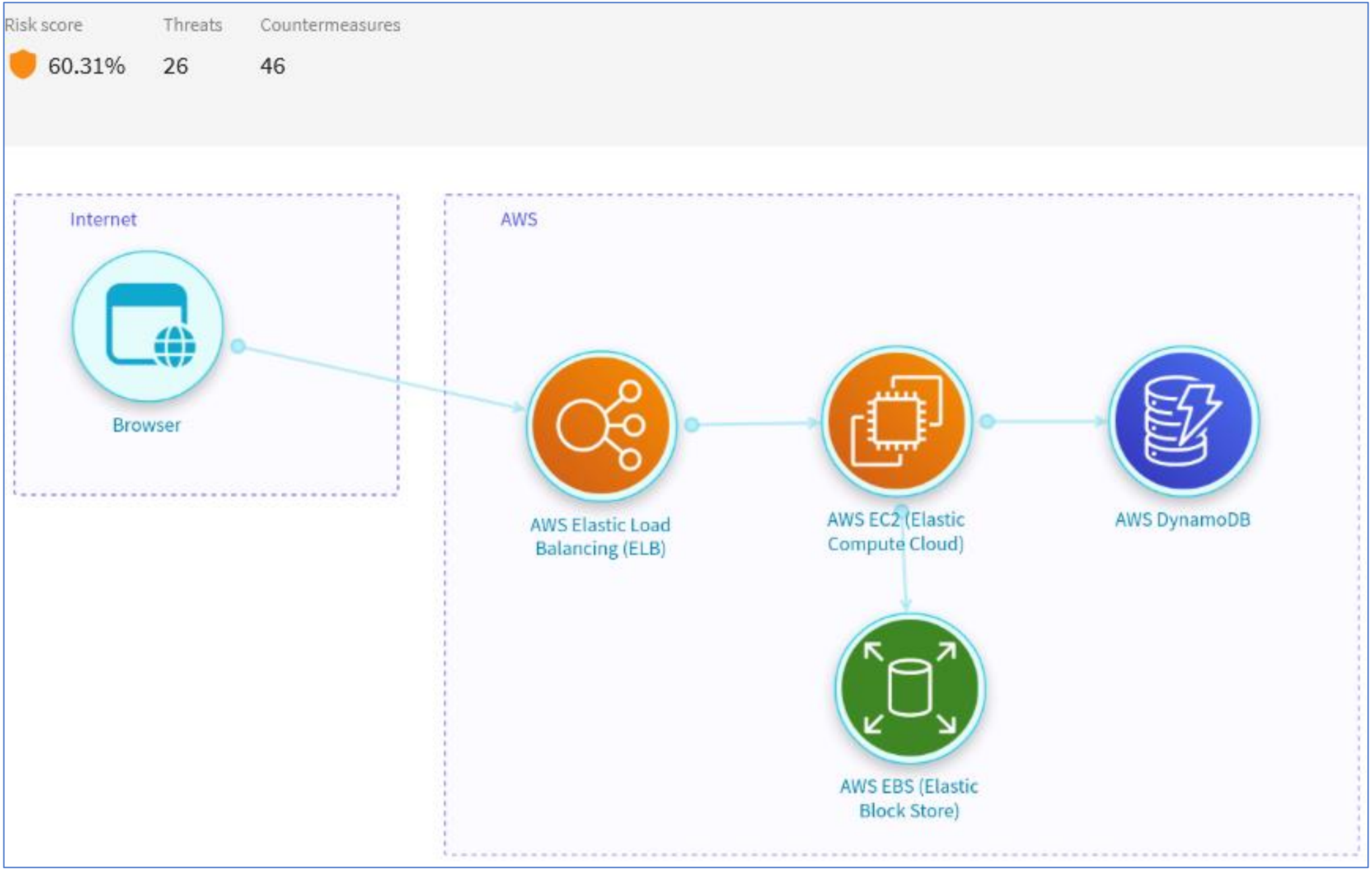
Architectural diagramming with draw.io

Limited technical and compliance reports

Receive free community updates

Collaborate with others in your Project

Risk score	Threats	Countermeasures
60.31%	26	46



Threats

26 threats found

- AWS DynamoDB 4
  - Elevation of Privilege 1
  - Information Disclosure 3
- AWS EBS (Elastic Block Store) 4
  - Information Disclosure 4
- AWS EC2 (Elastic Compute Cloud) 6
  - Denial of Service 2
  - Elevation of Privilege 2
  - Information Disclosure 2
- AWS Elastic Load Balancing (ELB) 7
  - Denial of Service 1
  - Elevation of Privilege 2
  - Information Disclosure 3
  - Spoofing 1
- Browser 5
  - Elevation of Privilege 1
  - Information Disclosure 1
  - Spoofing 2
  - Tampering 1

Top 10 high impact

Jeff AI Analysis Report

Countermeasures

⌃⌃	Browser	Deploy anti-phishing protection	Recommended
⌃⌃	Browser	Enforce strict certificate validation	Recommended
⌃⌃	AWS Elastic Load Bal…	Ensure encrypted communication for ELB listeners	Recommended
⌃	AWS Elastic Load Bal…	Enable Perfect Forward Secrecy (PFS) for ELB	Recommended
⌃	Browser	Implement client-side script blockers	Recommended
⌃	AWS EBS (Elastic Blo…	Encrypt all EBS snapshots to protect your data	Recommended
⌃	AWS EBS (Elastic Blo…	Set all new EBS volumes in your account to be encrypted by default	Recommended
⌃	AWS DynamoDB	Consider client-side encryption	Recommended
=	AWS DynamoDB	Use DynamoDB's encryption at rest to secure data stored in tables, backups, and streams	Recommended
=	AWS EC2 (Elastic Co…	Regular OS and application updates	Recommended

Easiest wins

Jeff AI Analysis Report

Countermeasures

⌃⌃	Browser	Configure automatic browser updates	Recommended
⌃⌃	Browser	Enforce strict certificate validation	Recommended
⌃⌃	Browser	Utilize encrypted communication tools	Recommended
⌃	Browser	Activate built-in browser security filters	Recommended
⌃	Browser	Implement client-side script blockers	Recommended

Critical threats with no mitigations

Jeff AI Analysis Report

10 threats found

Browser / Spoofing

Attackers conduct phishing attacks through deceptive websites

General Threat Description: Adversaries create deceptive websites that mimic legitimate ones to trick users into revealing sensitive information. Threat Agents/Attack Vectors: * Cybercriminals using social engineering techniques * Fake websites or compromised legitimate sites hosting phishing pages Impacts: * Credential theft and identity compromise * Unauthorized access to sensitive accounts or systems * Financial loss and reputational damage Example Attack Scenarios: * A user receives an email with a link to a fake banking website that looks identical to the real one, prompting them to enter their login details. * An attacker registers a domain similar to a…

Critical risk RO

Browser / Elevation of Privilege

Attackers exploit browser vulnerabilities to execute malicious code

General Threat Description: Adversaries leverage flaws in the browser's code to run unauthorized code, bypassing security measures and potentially compromising the entire system. Threat Agents/Attack Vectors: * Cybercriminals targeting known or zero-day browser vulnerabilities * Malicious websites and compromised ads delivering exploit code * Exploited browser extensions or plugins Impacts: * Unauthorized system access and control * Data theft or manipulation * Escalation of privileges on the host system Example Attack Scenarios: * An attacker uses a zero-day exploit on a popular browser via a compromised website, leading to malware installation. * A mali-…

Critical risk RO

Browser / Spoofing

Attackers intercept browser communications through man-in-the-middle (MitM) attacks

General Threat Description: Adversaries intercept and potentially alter communication between browsers and websites by exploiting insecure or misconfigured network protocols. Threat Agents/Attack Vectors: * Cybercriminals targeting unsecured Wi-Fi networks or misconfigured network devices * Attackers exploiting weak TLS/SSL configurations * Use of rogue access points or compromised routers Impacts: * Interception of sensitive data such as credentials and personal information * Data manipulation or session hijacking * Unauthorized access to private communications Example Attack Scenarios: * An attacker sets up a rogue Wi-Fi hotspot to capture unencrypte…

Critical risk RO

Browser / Tampering

Attackers distribute malware through compromised browser extensions

General Threat Description: Adversaries exploit or compromise browser extensions to distribute malware, leveraging the trust users place in these add-ons to execute malicious code within the browser environment. Threat Agents/Attack Vectors: * Cybercriminals submitting malicious extensions to official stores or hijacking updates of legitimate ones. * Social engineering tactics encouraging users to install unverified or counterfeit extensions. Impacts: * Unauthorized access to browser data and credentials * Installation of malware that may compromise the system * Potential lateral movement within a network through compromised systems Example Attack Scenario…

Critical risk RO

Browser / Information Disclosure

Attackers inject malicious scripts via cross-site scripting (XSS)

General Threat Description: Adversaries exploit vulnerabilities in web applications and browsers to inject malicious scripts, which then execute in users' browsers. Threat Agents/Attack Vectors: * Cybercriminals exploiting unvalidated input fields * Compromised or malicious websites hosting injected scripts Impacts: * Theft of session data and credentials * Unauthorized access to sensitive user information * Redirection to phishing or malicious sites Example Attack Scenarios: * An attacker injects a script into a forum post that steals users' cookies when viewed. * A vulnerable web form accepts unfiltered input, allowing an attacker to embed a script that executes upon…

High risk RO

AWS DynamoDB / Elevation of Privilege

Improperly configured IAM policies can lead to unauthorized data access

General Threat Description Improperly configured IAM (Identity and Access Management) policies in AWS can result in unauthorized access to DynamoDB tables. These policies govern who can access what resources and with what permissions. Misconfigurations can accidentally grant excessive privileges, allowing unauthorized users to read, write, or delete data, thereby compromising the security of the DynamoDB service. Threat Agents/Attack Vectors Threat agents include internal users such as employees or administrators who might misconfigure policies either accidentally or intentionally. External attackers can exploit stolen AWS credentials or use social engi-…

High risk RO

AWS DynamoDB / Information Disclosure

Without proper monitoring, security incidents can go undetected, leading to prolonged exposure

General Threat Description Without proper monitoring, security incidents involving AWS DynamoDB can go undetected, allowing malicious activities to persist and causing prolonged exposure to threats. Continuous monitoring is essential for detecting unauthorized access, suspicious activities, and potential security breaches in real-time. Lack of monitoring can result in delayed response to security incidents, increasing the risk of data compromise and operational disruptions. Threat Agents/Attack Vectors Threat agents include external attackers, malicious insiders, and automated threats such as malware. Attack vectors can involve exploiting vulnerabilities, using…

High risk RO

AWS DynamoDB / Information Disclosure

Data transmitted over the network can be intercepted by malicious actors

General Threat Description Data transmitted over a network can be intercepted by malicious actors using various techniques such as man-in-the-middle (MitM) attacks, packet sniffing, or session hijacking. In the context of AWS DynamoDB, data in transit between clients and the database service may be vulnerable if not adequately protected. Intercepted data can lead to exposure of sensitive information, unauthorized access, and manipulation of data. Threat Agents/Attack Vectors Threat agents include cybercriminals, hackers, or state-sponsored actors who intercept data using MitM attacks, deploying tools to capture network traffic, or exploiting vulnerabilities i…

High risk RO

AWS Elastic Load Balancing (ELB) / Spoofing

Weak authentication for ingress traffic

General Threat Description Weak authentication for ingress traffic to AWS Elastic Load Balancing (ELB) can allow unauthorized access, posing significant risks to security and data integrity. Without robust authentication, malicious actors can bypass security controls, leading to potential breaches. Threat Agents/Attack Vectors * Threat Agents: Hackers, malicious insiders, automated bots. * Attack Vectors: Exploiting weak or misconfigured authentication mechanisms, brute force attacks, credential stuffing. Impacts * Unauthorized Access: Malicious actors gain access to sensitive systems and data. * Data Breaches: Exposure of confidential information. * Service Disruption:…

High risk RO

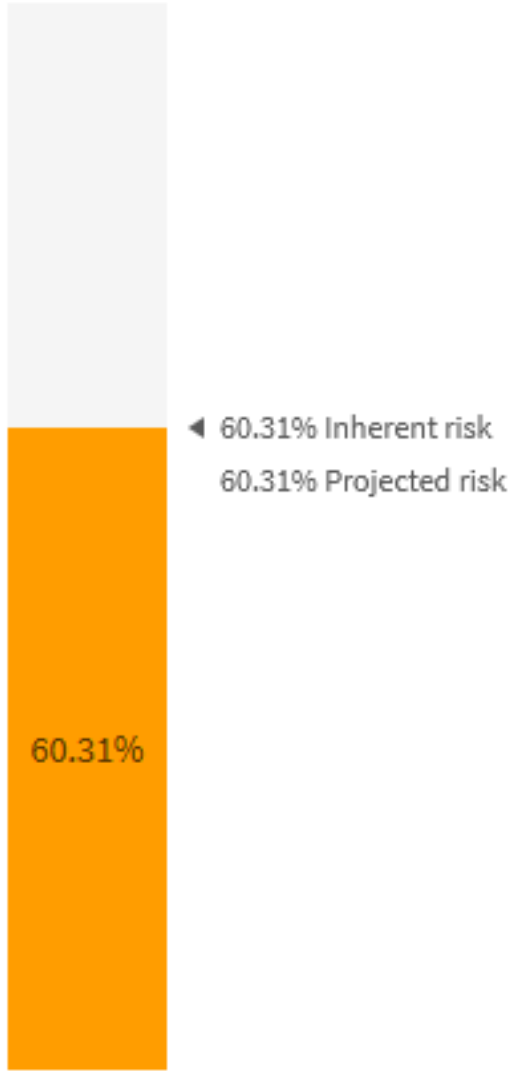
AWS DynamoDB / Information Disclosure

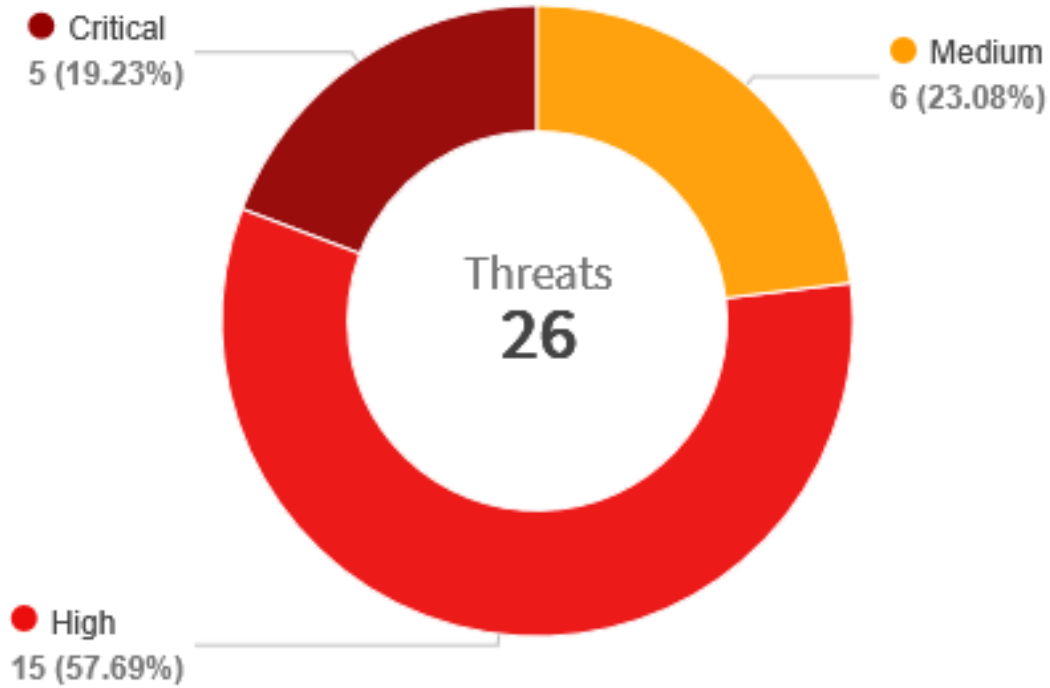
Sensitive data stored in DynamoDB can be exposed if not properly secured

General Threat Description Sensitive data stored in AWS DynamoDB can be exposed if not properly secured, leading to potential data breaches and unauthorized access. This can occur due to misconfigurations, inadequate access controls, lack of encryption, or insufficient monitoring. Ensuring that sensitive information such as personal, financial, or proprietary data is protected is critical to maintaining data confidentiality and integrity. Threat Agents/Attack Vectors Threat agents include malicious insiders with sufficient access privileges, external attackers exploiting misconfigured permissions or vulnerabilities, and automated scripts or applications with excessive access…

Medium risk RO

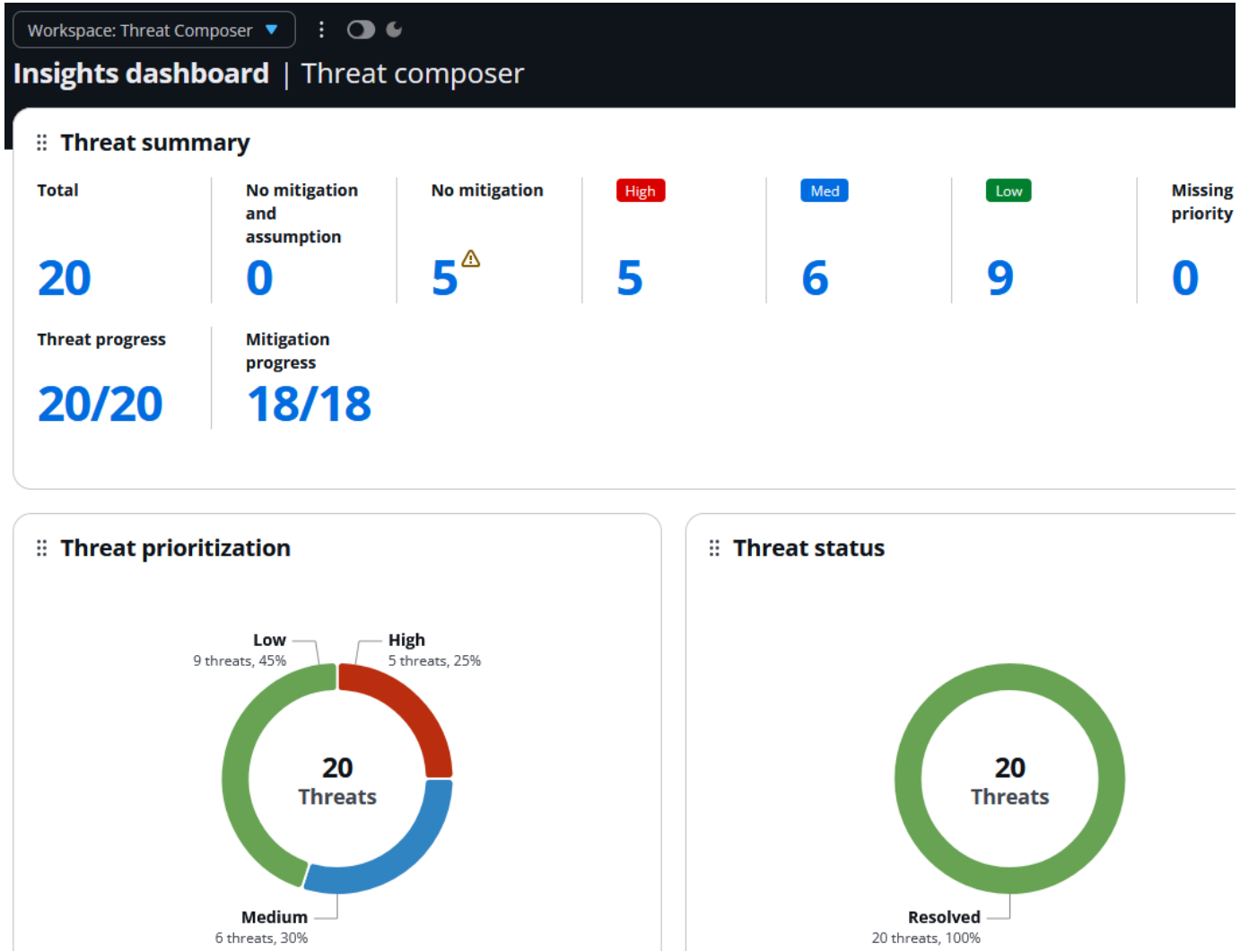
Threat risk distribution





AWS Threat Composer

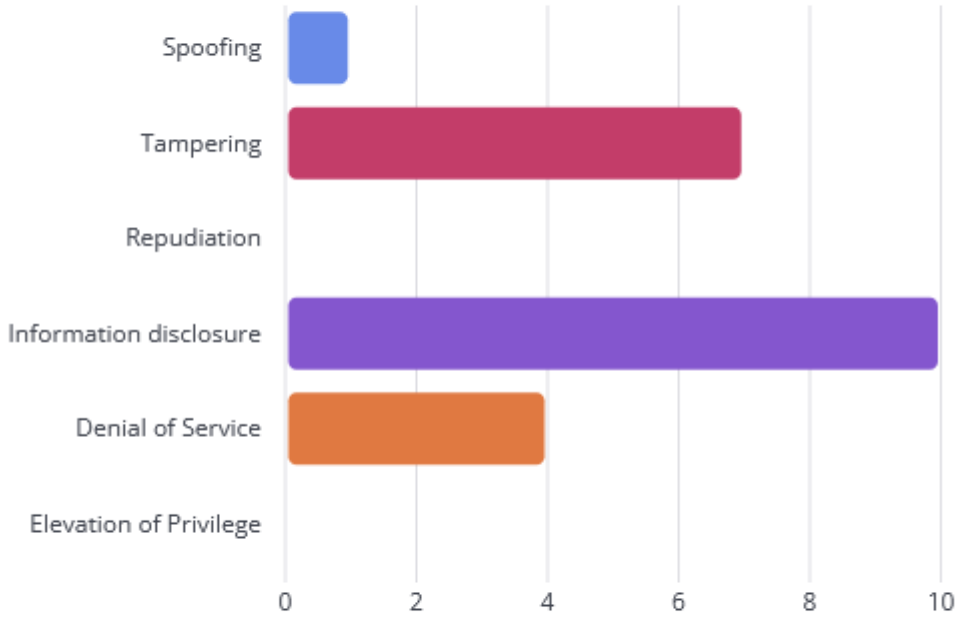
https://awslabs.github.io/threat-composer/workspaces/default/dashboard



Threat category distribution

Filter by threat priority

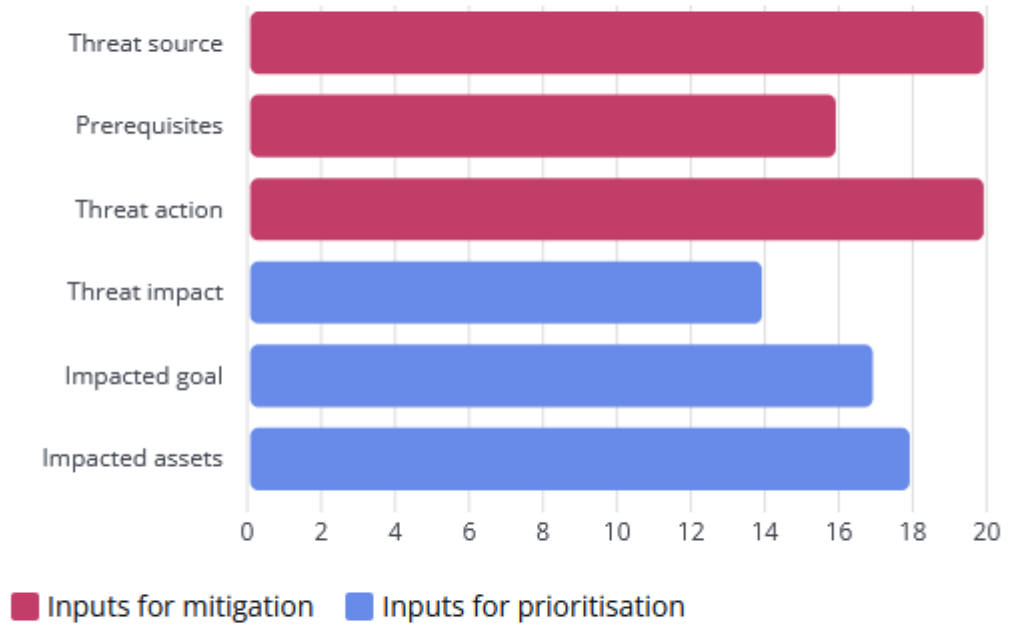
All



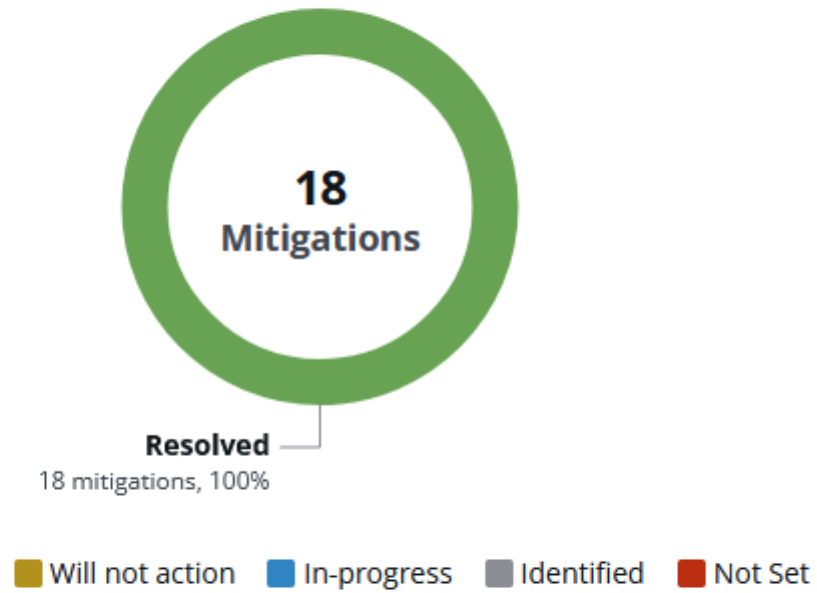
Threat grammar distribution

Filter by threat priority

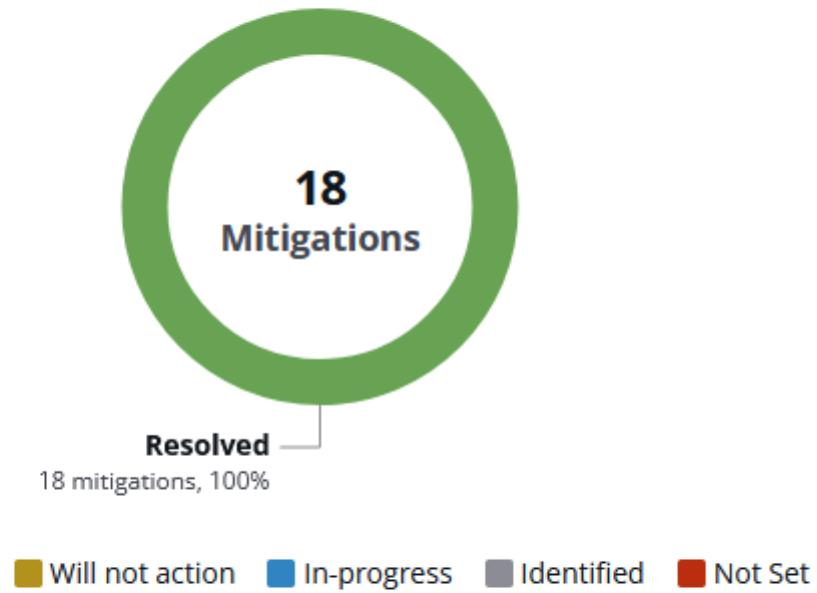
All



Mitigation status



Mitigation status



Features

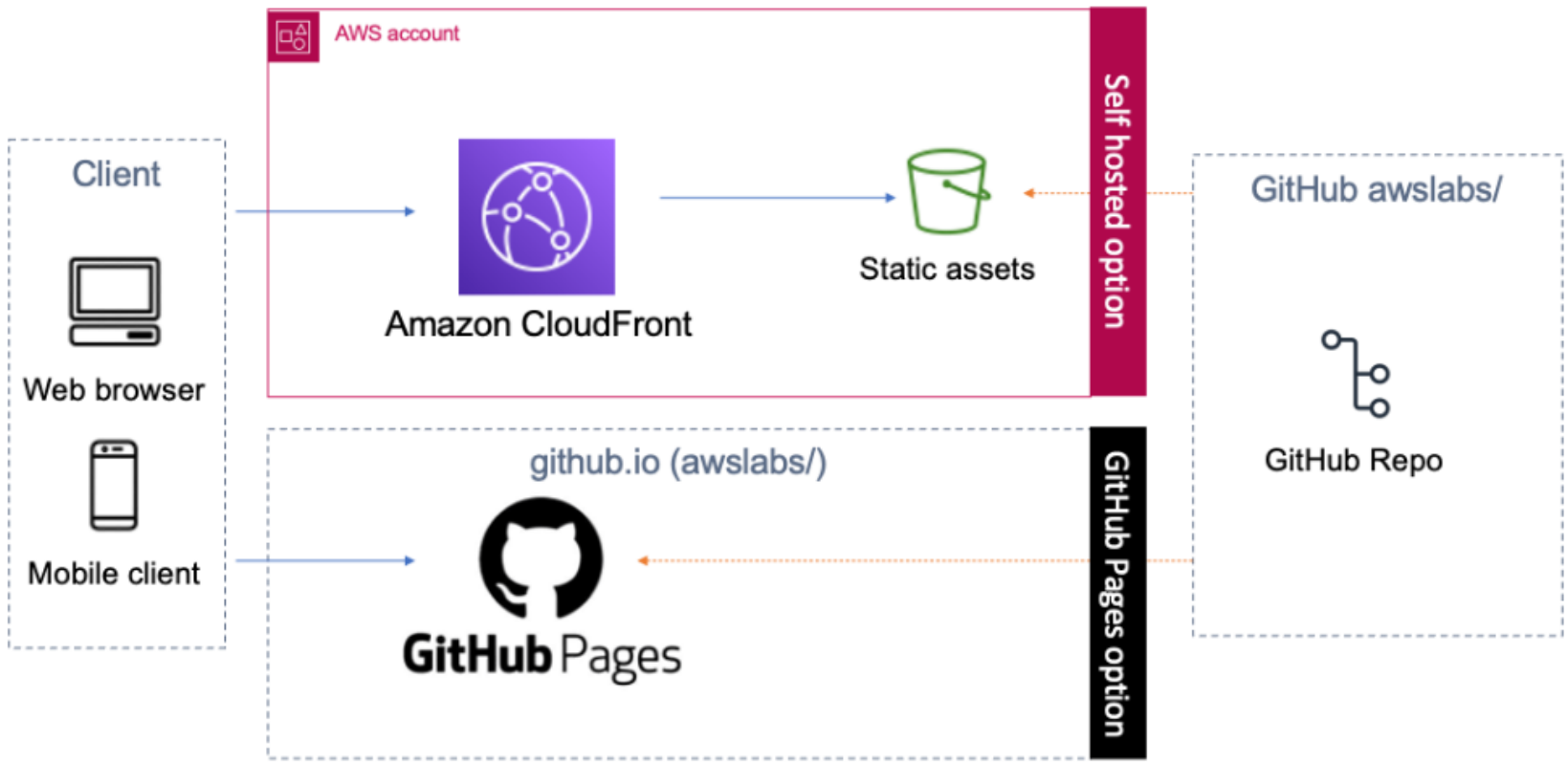
- Capture and store systems description, architecture diagram, and dataflow diagram.
- Capture and store assumptions related to the systems design, threats and/or mitigations, along with mapping of assumptions to threats to mitigations.
- Help iteratively compose useful threats, and encourage brainstorming. This feature is also available via a dedicated 'Threats Only' mode.
  - Rendering structured threat statements (aligned to a prescriptive threat grammar) based on user input.
  - Supporting an adaptive threat statement structure, this helps create progressively more complete threats.
  - Provide dynamic suggestions based on supplied and missing user input.
  - Provide complete threat statement examples to aid contextual brainstorming.
- Capture and store mitigation candidates and mapping to threats.
- Create a threat model document based on user-supplied input.
- Help users answer "Did we do a good enough job" by providing insights and suggestions for bar-raising actions via an 'Insights dashboard'
- Data persisted only client-side within the browser (100% local storage).
- JSON import/export capabilities to enable persistent storage, sharing, and version control outside of the web browser (e.g. by using git).
- Markdown and PDF static downloads of the threat model document.
- Workspace separation to allow working on multiple threat models.

There are two ways that a customer may interact with Threat Composer:

1. **Hosted** - Using the Github Pages hosted version. The source is hosted within the GitHub repo, and uses GitHub Actions for CI/CD to GitHub Pages.
2. **Self-hosted** - Deploying to their AWS account using the included AWS CDK app. Either:
    - **Static Website Only** - The application CloudFormation stack include a CloudFront distribution, S3 website bucket, and an associated AWS WAF WebACL.
    - **Static Website with CI/CD** - The CI/CD infrastructure includes a CodeCommit repository and a CodePipeline. The CodePipeline deploys the application stack (CloudFront distribution + S3 website bucket + AWS WAF WebACL) into the nominated dev and prod environments.

Notes:

- All user-supplied input is stored in browser storage, there is no backend nor any dynamic API calls
- By default all listed options **do not** include authentication to protect the static assets/
- By default the self-hosted options **do** have a restricted WebACL associated with the CloudFront distribution.



Threat 22

Resolved | Medium | Social engineering | Add tag

A threat actor that is able to trick a user into installing a malicous userscript extension (e.g. tampermonkey, browser extension) can read the contents of local browser storage, which leads to the exfiltration of the contents of browser storage to an endpoint controlled by the actor, resulting in reduced confidentiality of application metadata, threats, mitigations and assumptions

▶ **Linked mitigations (0)**

▶ **Linked assumptions (1)**

▶ **Metadata**

Threat 21

Resolved | Medium | XSS | Add tag

A threat actor that is able to target a user already using a benign userscript extension (e.g. tampermonkey) that integrates directly with local browser storage for quickly viewing a Threat Composer export can trick them into opening a malicious threat model that contains script tags (or similar), which leads to the exfiltration of the contents of browser storage via XSS due to the extension bypassing Threat Composers import validation and sanitisation protection, resulting in reduced confidentiality of application metadata, threats, mitigations and assumptions

▶ **Linked mitigations (3)**

▶ **Linked assumptions (0)**

▶ **Metadata**

Mitigation 19 Resolved

Add tag

Schema validation on data import

Linked threats (1)

Search threat

A threat actor that can trick a user into importing a JSON file can input an unexpected data schema, which leads to the user being unable to use the tool, until they clear the local data (or use a different browser)

Linked assumptions (0)

Metadata

Comments

B I Block type

Implementation in code - Import and data validation

Mitigation 18 Resolved

Add tag

Amazon S3 - Access logging

Linked threats (1)

Search threat

A threat actor with write access to the objects hosted on the static asset S3 Bucket can modify the code, which leads to exfiltration of user-supplied input to an attacker controlled endpoint, resulting in reduced confidentiality of threats, mitigations, assumptions and application metadata

Linked assumptions (0)

Metadata

Comments

B I Block type

Implementation in code - PDK Static Website construct - S3 access logging

Assumptions

Assumption Number	Assumption	Linked Threats	Linked Mitigations	Comments
A-0005	Security awareness training is the most effective mitigation against social engineering attacks, and one cannot rely solely on technical mitigations	T-0006: A threat actor with possession of a similar domain name can trick our users into interacting with an illegitimate endpoint, resulting in reduced confidentiality of threats, mitigations, assumptions and application metadata		A consuming customer should have a security awareness program to help to educate their users on how to reuce the likelihood of being socially engineered. See article Amazon releases free cybersecurity awareness training
A-0004	Customer deploying solution will follow AWS Well-Architected best practices	T-0005: An external threat actor who has a sufficiently privileged IAM Principal in the AWS account can modify the configuration of the CloudFront distribution, which leads to the distribution serving content from an origin that is unexpected or contains malicious content, resulting in reduced integrity of code running in the user's browser and application configuration T-0004: A threat actor with write access to the objects hosted on the static asset S3 Bucket can modify the code, which leads to exfiltration of user-supplied input to an attacker controlled endpoint, resulting in reduced confidentiality of threats, mitigations, assumptions and application metadata T-0016: A valid user who has forked and modified the source code to include their own data (e.g. additional example threat statements) can deploy Threat composer without network restrictions or authentication, which leads to discovery of the additional data by an adversary, resulting in reduced confidentiality		AWS Well-Architected documentation

Summary and Conclusions

Summary

- ***Common Pitfalls in Cloud Threat Modelling***
  - Treating cloud like a traditional datacentre.
  - Misunderstanding the cloud provider's vs your responsibility.
  - Lack of clear ownership across layers and teams.
  - Skipping threat modelling for pipelines and third-party SaaS dependencies.

- ***Conclusion and Takeaways***
  - Threat modelling must extend across all cloud layers: infra, platform, application.
    - Focus on reusable IaC components and design of guardrails – Shift Left, Left
  - Consider SaaS, CI/CD, IaC, and ephemeral runtime components.
  - Make it collaborative, continuous, and tool-assisted.
  - Start small, iterate fast — threat modelling doesn't need to be perfect to be valuable

Benefits of Threat Modelling

- Helps identify and prioritise threats, early in the lifecycle
  - Helping to optimise resources and limited budgets
  - Reducing risk exposure
  - Assessing reusable IaC components can deliver large gains
- Considers evolving threat landscape
- Helps developers design and build secure software.
- Develops security skills/mindset within project/engineering teams
- Encourages collaboration on security initiatives

Making IT
good for society

BCS DevSecOps Group

https://www.bcs.org/membership-and-registrations/member-communities/devsecops-specialist-group/

Roy Harrow - chairdevsecops@bcs.org


Q + A