Husky AI

Owner:
Reviewer:
Contributors: Imported from TM-BOM
Date Generated: Tue Oct 07 2025

Executive Summary

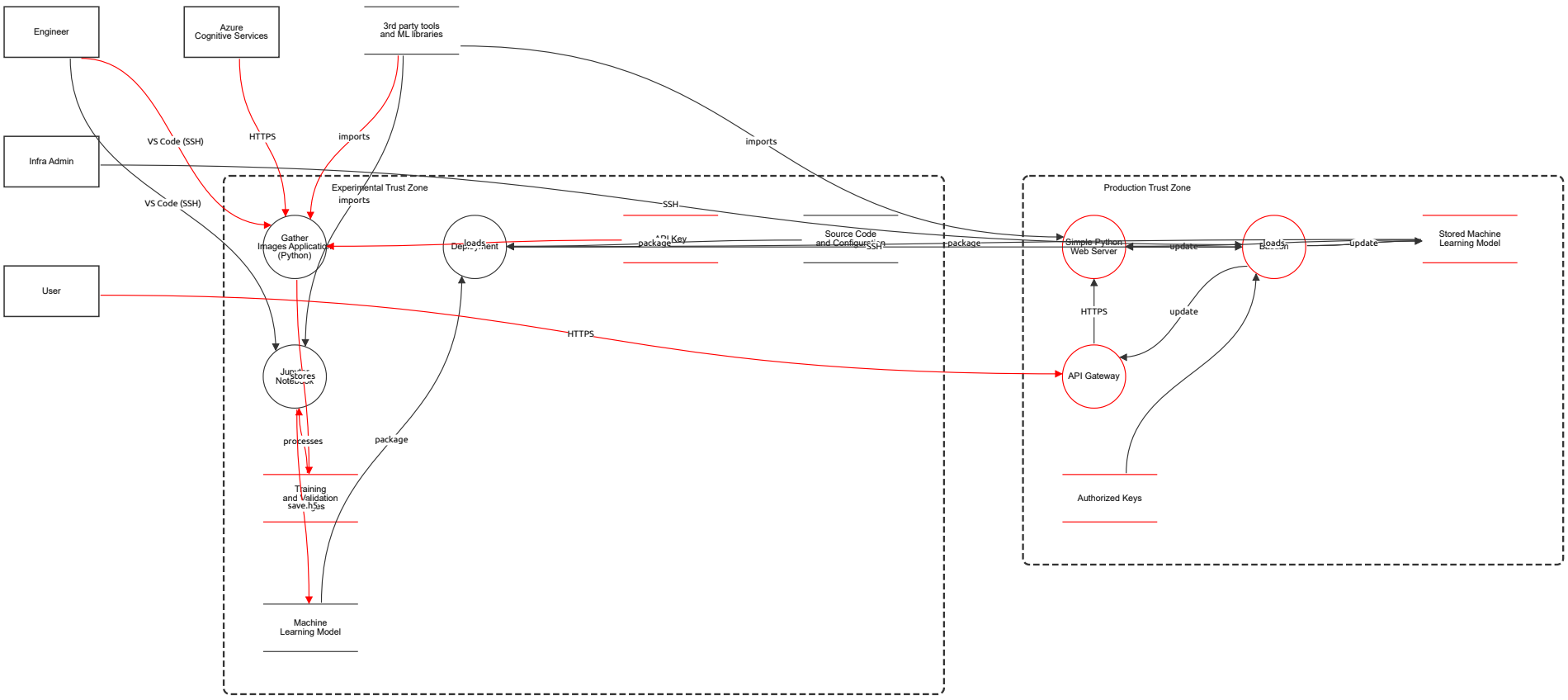
High level system description

A machine learning system to classify Huskies vs dogs. HuskyAI is a machine learning system designed to classify images and distinguish between huskies and non-huskies. It integrates secure data handling practices with a robust convolutional neural network (CNN) for image recognition. Secure Image Retrieval: HuskyAI uses TLS to securely fetch images from Azure Cognitive Services, ensuring encryption during data transmission and validating the server's authenticity to prevent man-in-the-middle attacks. Data Storage and Access Controls: Azure Blob Storage is used to store datasets, with public access fully blocked. Access is controlled using Role-Based Access Control (rbac) and Attribute-Based Access Control (ABAC) to enforce granular, identity-based permissions. Jupyter Notebooks, which host model development and experimentation, are also secured with rbac and ABAC, preventing unauthorized public access. Developer Authentication: Developers access the system through SSH keys protected by passphrases. This adds an additional layer of security, reducing the likelihood of unauthorized access even if keys are exposed. Model and Dataset Dataset Composition: The dataset comprises approximately 1,300 husky images and 3,000 non-husky images sourced via Bing's image search. Data undergoes manual cleansing and is split into training and validation sets to enhance model performance. Model Design: HuskyAI employs a CNN with: Convolutional layers for feature extraction. Max-pooling layers for dimensionality reduction. Dropout layers to prevent overfitting. Dense layers for final classification. The model is trained with the Adam optimizer and a learning rate of 0.0005, optimized for accuracy and computational efficiency. Security Considerations rbac and ABAC controls across storage and development environments ensure sensitive data and configurations are protected. TLS ensures secure communication channels, preventing eavesdropping or data interception during image retrieval. Applications HuskyAI is tailored for accurate image classification and can be adapted for other domains requiring precise visual differentiation, with a focus on maintaining strong security postures. HuskyAI combines state-of-the-art machine learning techniques with stringent security controls, including secure communications, robust access management, and encrypted developer authentication, to deliver a reliable and secure image classification system.

Summary

Total Threats	19
Total Mitigated	0
Total Open	19
Open / Critical Severity	0
Open / High Severity	9
Open / Medium Severity	8
Open / Low Severity	2

Husky AI


Engineer
Azure Cognitive Services
3rd party tools and ML libraries
Infra Admin
User
VS Code (SSH)
HTTPS
imports
SSH
Experimental Trust Zone
Gather Images Application (Python)
Deployment
Source Code and Configuration
API Key
package
Jupyter Notebook
stores
processes
Training and validation save.h5as
Machine Learning Model
Production Trust Zone
Simple Python Web Server
loads
update
Stored Machine Learning Model
API Gateway
Authorized Keys

Husky AI

Engineer (Actor)

Description: A Data Engineer responsible for building, training, and deploying machine learning models.

Number	Title	Type	Severity	Status	Score	Description	Mitigations

Infra Admin (Actor)

Description: Administrator responsible for securing and maintaining production infrastructure.

Number	Title	Type	Severity	Status	Score	Description	Mitigations

Azure Cognitive Services (Actor)

Description: External service providing resources for machine learning experimentation.

Number	Title	Type	Severity	Status	Score	Description	Mitigations

User (Actor)

Description: External user interacting with the HuskyAI system via the API Gateway.

Number	Title	Type	Severity	Status	Score	Description	Mitigations

3rd party tools and ML libraries (Store)

Description: External third party tools for the services

Number	Title	Type	Severity	Status	Score	Description	Mitigations

Gather Images Application (Python) (Process)

Description: This is a Python-based application responsible for gathering images from external sources, specifically Azure Cognitive Services, and storing them in the designated Training and Validation Images storage.

Number	Title	Type	Severity	Status	Score	Description	Mitigations

Jupyter Notebook (Process)

Description: A Jupyter Notebook environment that processes the images stored in Training and Validation Images, executes code using external ML libraries, and provides a UI for engineers to interact with and manipulate data, allowing for iterative model development. It can save trained machine learning models to Machine Learning Model storage.

Number	Title	Type	Severity	Status	Score	Description	Mitigations

Deployment (Process)

Description: Handles the deployment of the machine learning model by packaging the model and all necessary source code and configuration stored in Source Code and Configuration. It receives the final model from Jupyter Notebook and prepares it for deployment to the production environment.

Number	Title	Type	Severity	Status	Score	Description	Mitigations

Training and Validation Images (Store)

Description: Contains images used for training and validation of machine learning models.
Data set: Training and Validation Images
Contains images used for training and validation of machine learning models.
Record count maximum of 100000 with data sensitivity of biz and access control methods of rbac

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Model poisoning via tampered images	Tampering	High	Open		Training and Validation Images store (inside Experimental Trust Zone, data sensitivity: biz) holds ~4300 images. If accessed or modified by unauthorized users (despite RBAC), tampered images could poison the ML model, leading to inaccurate classifications.	Implement data validation pipelines to detect anomalies in images. Use immutable storage and audit all access to the blob.

API Key (Store)

Description: Stores API keys for secure access to external services.
Data set: API Keys
Stores API keys for secure access to external services.
Record count maximum of 20 with data sensitivity of cred and access control methods of rbac

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Unauthorized access to API keys	Information Disclosure	High	Open		API Key store (inside Experimental Trust Zone) holds sensitive credentials (data sensitivity: cred) protected by RBAC. If access controls are misconfigured or bypassed, keys could be disclosed, compromising external service access.	Use Azure Key Vault for credential management with least-privilege RBAC and ABAC. Enable auditing and automatic key rotation.

Machine Learning Model (Store)

Description: Contains the machine learning models in serialized format.
Data set: Bastion Logs
Contains trained machine learning models in serialized format for production use.
Record count maximum of 5000 with data sensitivity of biz and access control methods of acl

Number	Title	Type	Severity	Status	Score	Description	Mitigations

Source Code and Configuration (Store)

Description: Stores source code and configuration files for deployment and production setup.
Data set: Source Code and Configuration
Stores source code and configuration files for deployment and production setup.
Record count maximum of 200 with data sensitivity of biz and access control methods of rbac

Number	Title	Type	Severity	Status	Score	Description	Mitigations

Simple Python Web Server (Process)

Description: Serves as simple web server

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Denial of Service on web server	Denial of Service	Medium	Open		Simple Python Web Server (inside Production Trust Zone) receives routed requests from API Gateway and direct Bastion updates. Internal or proxied floods could overwhelm the server, disrupting image classification.	Implement resource limits in the Python server, use load balancers, and monitor for abuse patterns.
	Information Disclosure via error messages	Information Disclosure	Low	Open		The web server processes ML inferences; verbose error responses could disclose internal details like model versions or paths, aiding attackers.	Sanitize error messages to avoid leaking sensitive information. Use generic responses for production errors.

API Gateway (Process)

Description: Serves as the entry point for external users to interact with the production environment via HTTPS. It routes user requests to the Simple Python Web Server and ensures secure communication. The API Gateway enforces request validation and manages APIs exposed to the public while ensuring access control to internal services.

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Repudiation of API actions	Repudiation	Medium	Open		API Gateway (inside Production Trust Zone) handles external HTTPS requests from User. Without proper logging, users could deny performing actions like model inferences, complicating accountability.	Enable comprehensive logging of all API requests with user identifiers (e.g., JWT claims) and timestamps. Use non-repudiable auth mechanisms.

Bastion (Process)

Description: A secure access management component for administrative functions. It provides controlled SSH access for the Infrastructure Admin to internal production resources, such as the Stored Machine Learning Model and Simple Python Web Server.

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Elevation of privilege through Bastion compromise	Elevation of Privilege	High	Open		Bastion process (inside Production Trust Zone) provides SSH access to production resources like Simple Python Web Server and Stored Machine Learning Model. If compromised (e.g., via weak SSH), attackers gain elevated access to the entire zone.	Harden Bastion with minimal services, use bastion hosts with MFA, and restrict outbound connections. Regularly patch and audit.

Authorized Keys (Store)

Description: Contains SSH keys used for securing administrative access.
Data set: Authorized Keys
Contains SSH keys used for securing administrative access.
Record count maximum of 100 with data sensitivity of cred and access control methods of rbac

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Compromise of SSH authorized keys	Elevation of Privilege	High	Open		Authorized Keys store (inside Production Trust Zone) contains SSH keys (data sensitivity: cred). Compromise could allow elevation of privilege, enabling unauthorized admin access to production resources via Bastion.	Store keys in a secure vault with passphrase protection. Implement just-in-time access and monitor for key usage anomalies.

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Information Disclosure of SSH keys	Information Disclosure	High	Open		The store is encrypted at rest, but if RBAC fails or during access, keys could be exposed, leading to unauthorized SSH access across production zone.	Enforce strict RBAC/ABAC, encrypt keys at rest and in transit, and use hardware security modules (HSM) for key storage.

Stored Machine Learning Model (Store)

Description: Contains storage for machine learning models in serialized format.
Data set: Stored Machine Learning Models
Contains trained machine learning models in serialized format for production use.
Record count maximum of 10 with data sensitivity of biz and access control methods of rbac

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Tampering with production models	Tampering	High	Open		Stored Machine Learning Model store (inside Production Trust Zone, data sensitivity: biz) holds serialized models. Unauthorized modifications via Bastion access could tamper with models, causing faulty classifications in production.	Use version control and signing for models. Implement access logging and integrity checks on load.

HTTPS (Data Flow)

Description: Transfer data from Azure Cognitive Services to Gather Images Application in Python.

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Tampering with fetched images	Tampering	Medium	Open		The flow 'HTTPS' from Azure Cognitive Services (external actor outside any trust boundary) to Gather Images Application (Python) (process inside Experimental Trust Zone) crosses into a trusted zone. Although encrypted with HTTPS, images from an external source may contain tampered or malicious content if not validated upon receipt, potentially poisoning the training dataset.	Implement content validation and integrity checks (e.g., checksums or digital signatures) on received images before storage. Use certificate pinning for the HTTPS connection to Azure Cognitive Services.

imports (Data Flow)

Description: Transfer data from Third Party tools and ML libraries to Gather Images Application in Python.

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Supply chain tampering via libraries	Tampering	High	Open		The flow 'imports' from 3rd party tools and ML libraries (store outside Experimental Trust Zone due to position) to Gather Images Application (Python) (process inside Experimental Trust Zone) introduces external dependencies into the trusted zone. Malicious or compromised third-party libraries could tamper with the image gathering process or inject vulnerabilities.	Scan all third-party libraries for vulnerabilities using tools like Dependabot or Snyk. Implement runtime integrity checks and sandbox the import process.

imports (Data Flow)

Description: Transfer data from Third Party tools and ML libraries to Jupyter Notebook.

Number	Title	Type	Severity	Status	Score	Description	Mitigations

VS Code (SSH) (Data Flow)

Description: Transfer data from Engineer to Gather Images Application in Python.

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Spoofing engineer identity	Spoofing	Medium	Open		The flow 'VS Code (SSH)' from Engineer (external actor outside trust boundaries) to Gather Images Application (Python) (process inside Experimental Trust Zone) relies on SSH for ingress. An attacker could spoof the engineer's identity if SSH keys are compromised, gaining unauthorized access to the development environment.	Enforce multi-factor authentication (MFA) for SSH access, use key passphrases, and rotate keys regularly. Monitor SSH logs for anomalous access.

VS Code (SSH) (Data Flow)

Description: Transfer code and ML models from Engineer locally to Jupyter Notebook.

Number	Title	Type	Severity	Status	Score	Description	Mitigations

stores (Data Flow)

Description: Transfer images from Gather Images Application to Training and Validation Images.

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Tampering with unencrypted image storage	Tampering	Medium	Open		The flow 'stores' from Gather Images Application (Python) (process inside Experimental Trust Zone) to Training and Validation Images (store inside Experimental Trust Zone) is not encrypted (isEncrypted: false). Internal adversaries or network attackers within the zone could tamper with images in transit, leading to corrupted training data.	Enable TLS encryption for all internal data flows between processes and stores. Use message authentication codes (MAC) to ensure integrity.
	Information Disclosure of training images	Information Disclosure	Low	Open		The flow 'stores' from Gather Images Application (Python) to Training and Validation Images is unencrypted and occurs within the Experimental Trust Zone. Although internal, shoulder-surfing or compromised internal network could disclose business-sensitive image data.	Encrypt all internal communications with TLS 1.2+. Implement network segmentation to isolate experimental zone traffic.

loads (Data Flow)

Description: API Key Storage to Gather Images Application in Python.

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Information Disclosure of API keys in transit	Information Disclosure	High	Open		The flow 'loads' from API Key (store inside Experimental Trust Zone) to Gather Images Application (Python) (process inside Experimental Trust Zone) is not encrypted (isEncrypted: false). Sensitive credentials could be intercepted by internal attackers, enabling unauthorized access to external services.	Encrypt credential flows using TLS or secure vaults like Azure Key Vault with just-in-time access. Avoid direct loading; use token-based auth.

processes (Data Flow)

Description: Load from Training and Validation Images to Jupyter Notebook.

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Tampering with training data loading	Tampering	Medium	Open		The flow 'processes' from Training and Validation Images (store inside Experimental Trust Zone) to Jupyter Notebook (process inside Experimental Trust Zone) is unencrypted. Attackers with internal access could tamper with loaded images, affecting model training integrity.	Implement end-to-end encryption for data loads and validate data integrity upon receipt in Jupyter Notebook.

package (Data Flow)

Description: Transfer data from Machine Learning Model to Deployment.

Number	Title	Type	Severity	Status	Score	Description	Mitigations

save.h5 (Data Flow)

Description: Transfer final model from Jupyter Notebook to Machine Learning Model.

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Information Disclosure of trained models	Information Disclosure	Medium	Open		The flow 'save.h5' from Jupyter Notebook (process inside Experimental Trust Zone) to Machine Learning Model (store inside Experimental Trust Zone) is unencrypted. Business-sensitive model artifacts could be disclosed if the internal network is compromised.	Encrypt model serialization and storage flows with TLS. Store models in encrypted format at rest.

package (Data Flow)

Description: Transfer from Machine Learning Model Blob to Deployment Service.

Number	Title	Type	Severity	Status	Score	Description	Mitigations

package (Data Flow)

Description: Transfer data from Source Code and Configuration to Deployment.

Number	Title	Type	Severity	Status	Score	Description	Mitigations

HTTPS (Data Flow)

Description: Transfer from User to API Gateway.

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Denial of Service on API entry point	Denial of Service	High	Open		The flow 'HTTPS' from User (external actor outside trust boundaries) to API Gateway (process inside Production Trust Zone) represents public ingress. External attackers could flood the gateway with requests, causing denial of service to the production system.	Deploy rate limiting, CAPTCHA for suspicious traffic, and a Web Application Firewall (WAF) at the API Gateway. Use auto-scaling for load.
	Spoofing user requests	Spoofing	Medium	Open		The ingress flow 'HTTPS' from User to API Gateway crosses from external to Production Trust Zone. Without strong authentication, attackers could spoof legitimate user requests to access the classification API.	Implement JWT or API key authentication with validation at the gateway. Enforce origin checks and mTLS where possible.

update (Data Flow)

Description: Transfer data from Bastion to API Gateway.

Number	Title	Type	Severity	Status	Score	Description	Mitigations

HTTPS (Data Flow)

Description: Transfer data from API Gateway to Simple Python Web Server.

Number	Title	Type	Severity	Status	Score	Description	Mitigations

update (Data Flow)

Description: Transfer data from Bastion to Simple Python Web Server.

Number	Title	Type	Severity	Status	Score	Description	Mitigations

loads (Data Flow)

Description: Transfer sensitive data from Stored Machine Learning Model to Simple Python Web Server.

Number	Title	Type	Severity	Status	Score	Description	Mitigations

SSH (Data Flow)

Description: Transfer sensitive data from Deployment Service to Bastion

Number	Title	Type	Severity	Status	Score	Description	Mitigations

update (Data Flow)

Description: Transfer sensitive data from Bastion to Stored Machine Learning Model.

Number	Title	Type	Severity	Status	Score	Description	Mitigations

SSH (Data Flow)

Description: Transfer data from Infrastructure Admin to Bastion.

Number	Title	Type	Severity	Status	Score	Description	Mitigations

update (Data Flow)

Description: Transfer sensitive data from Bastion to Stored Machine Learning Model.

Number	Title	Type	Severity	Status	Score	Description	Mitigations

(Data Flow)

Description: Transfer sensitive data from Authorized Keys Storage to Bastion.

Number	Title	Type	Severity	Status	Score	Description	Mitigations

imports (Data Flow)

Description: Transfer data from Third Party tools and ML libraries to Simple Python Web Server.

Number	Title	Type	Severity	Status	Score	Description	Mitigations