Husky AI

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

Executive Summary

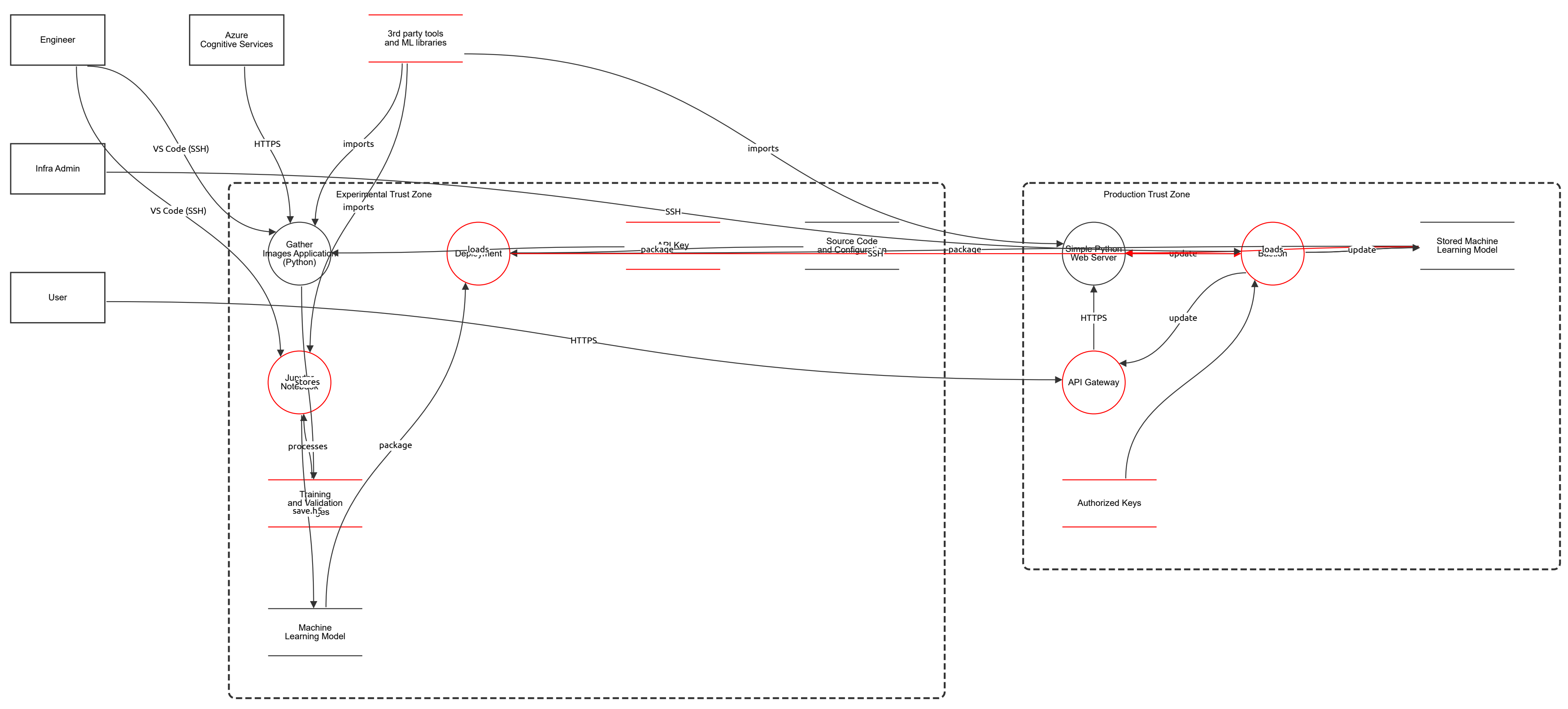
High level system description

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	10
Total Mitigated	0
Total Open	10
Open / Critical Severity	0
Open / High Severity	7
Open / Medium Severity	3
Open / Low Severity	0

Husky AI


Engineer
Azure Cognitive Services
3rd party tools and ML libraries
Infra Admin
User
VS Code (SSH)
HTTPS
imports
imports
VS Code (SSH)
imports
Experimental Trust Zone
SSH
Gather Images Application (Python)
loads
Deployment
package
API Key
Source Code and Configuration
SSH
package
Jupyter Notebook
stores
HTTPS
processes
package
Training and Validation Images
save.h5
Machine Learning Model
Production Trust Zone
Simple Python Web Server
update
loads
Bastion
update
Stored Machine Learning Model
HTTPS
update
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
	Supply Chain Attack via Malicious ML Libraries	Tampering	High	Open		The '3rd party tools and ML libraries' store provides dependencies to multiple processes like 'Jupyter Notebook' and 'Simple Python Web Server'. A compromised or malicious library could be introduced, leading to code execution, data tampering, or model poisoning within both the experimental and production environments.	Use a dependency scanner (e.g., Snyk, Dependabot) to identify known vulnerabilities. Vet all third-party libraries before use. Pin dependency versions in requirements files. Consider hosting a private repository for approved packages to control the source of dependencies.

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
	Arbitrary Code Execution in Jupyter Notebook	Elevation of Privilege	High	Open		The 'Jupyter Notebook' process is a common target for vulnerabilities. An attacker exploiting a weakness in the notebook server or a library it uses could execute arbitrary code within the 'Experimental Trust Zone', gaining access to training data, models, and potentially pivoting to other systems.	Run the Jupyter server with minimal privileges in a containerized environment. Apply security patches promptly. Require strong authentication for notebook access. Do not expose the notebook server directly to the internet. Regularly scan the environment for vulnerabilities.

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
	Compromise of Deployment Service leading to Production Access	Elevation of Privilege	High	Open		The 'Deployment' process initiates an SSH connection to the 'Bastion' in the 'Production Trust Zone'. If this service is compromised, an attacker can use its credentials to pivot from the less-secure 'Experimental Trust Zone' to the highly-sensitive production environment, bypassing security controls.	Ensure the deployment service runs with the principle of least privilege. Use short-lived, single-use credentials for deployment. Implement strict network policies (e.g., security groups) to limit communication from the experimental to the production zone. Heavily monitor the SSH connection from this service for anomalous behavior.

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 Skewing via Training Data Poisoning	Tampering	High	Open		An attacker with access to the 'Training and Validation Images' data store could maliciously modify, add, or delete images. This data poisoning can compromise the integrity of the trained ML model, causing it to misclassify images, exhibit biased behavior, or contain backdoors that can be exploited in production.	Implement strict access controls (RBAC) and versioning on the data store. Generate and verify checksums for the dataset before each training run. Use data validation and anomaly detection techniques to identify potentially poisoned data points.

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
	Credential Leakage from API Key Storage	Information Disclosure	High	Open		The 'API Key' store contains sensitive credentials. Improper access controls or a vulnerability in a process that reads from it (like 'Gather Images Application') could lead to the disclosure of these keys, allowing an attacker to abuse the 'Azure Cognitive Services' account.	Use a dedicated secrets management solution like Azure Key Vault or HashiCorp Vault. Enforce strict, minimal access policies (RBAC) to the secrets. Rotate keys regularly. Ensure applications retrieve secrets at runtime rather than storing them in configuration files.

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

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
	Denial of Service via Resource Exhaustion	Denial of Service	Medium	Open		The 'API Gateway' is exposed to the external 'User'. An attacker can send a high volume of requests or computationally expensive requests (e.g., large image uploads) to overwhelm the gateway or the backend 'Simple Python Web Server', leading to a denial of service for legitimate users.	Implement rate limiting and request throttling at the 'API Gateway'. Use a Web Application Firewall (WAF) to block malicious traffic patterns. Implement input validation to reject overly large or malformed requests.

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
	Insufficient Logging of Administrative Actions	Repudiation	Medium	Open		The 'Bastion' process is the single point of administrative access to the 'Production Trust Zone'. Without comprehensive and immutable logging of all commands executed through the bastion, a malicious actor could perform unauthorized actions without leaving a clear audit trail, making it impossible to determine what changes were made.	Implement detailed session logging for all SSH connections through the bastion. Forward logs to a separate, secure logging system (e.g., a SIEM) with append-only permissions. Set up alerts for suspicious commands or activities.

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
	Unauthorized Modification of Authorized Keys	Tampering	High	Open		The 'Authorized Keys' data store controls administrative access to the 'Production Trust Zone' via the 'Bastion'. An attacker who gains write access to this store could add their own SSH public key, granting themselves persistent, privileged access to the production environment.	Implement strict Role-Based Access Control (RBAC) on the key store. Use file integrity monitoring to detect any changes to the authorized keys file. Regularly audit the keys present in the store to ensure only authorized administrators have access.

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

HTTPS (Data Flow)

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

Number	Title	Type	Severity	Status	Score	Description	Mitigations

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

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

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

loads (Data Flow)

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

Number	Title	Type	Severity	Status	Score	Description	Mitigations

processes (Data Flow)

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

Number	Title	Type	Severity	Status	Score	Description	Mitigations

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

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

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
	Model Theft via Unencrypted Internal Traffic	Information Disclosure	Medium	Open		The 'loads' data flow from the 'Stored Machine Learning Model' to the 'Simple Python Web Server' is not marked as encrypted. An attacker who has gained a foothold within the 'Production Trust Zone' network could sniff this traffic and steal the proprietary machine learning model.	Enforce TLS for all internal communications, even within a trusted zone (zero-trust networking). Use a service mesh (e.g., Istio, Linkerd) to automatically enforce mTLS between all services.

SSH (Data Flow)

Description: Transfer sensitive data from Deployment Service to Bastion

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Privilege Escalation from Experimental to Production Zone	Elevation of Privilege	High	Open		The SSH flow from the 'Deployment' service in the 'Experimental Trust Zone' to the 'Bastion' in the 'Production Trust Zone' represents a critical trust boundary crossing. A compromise of the deployment service or its credentials would allow an attacker to directly pivot into the production environment.	Use ephemeral, single-use credentials for the deployment SSH connection. Strictly limit the commands that can be executed by the deployment key on the bastion host. Implement strong network segmentation and firewall rules to restrict all other traffic between the two zones. Monitor this specific flow for any anomalous activity.

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