Husky AI

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

Executive Summary

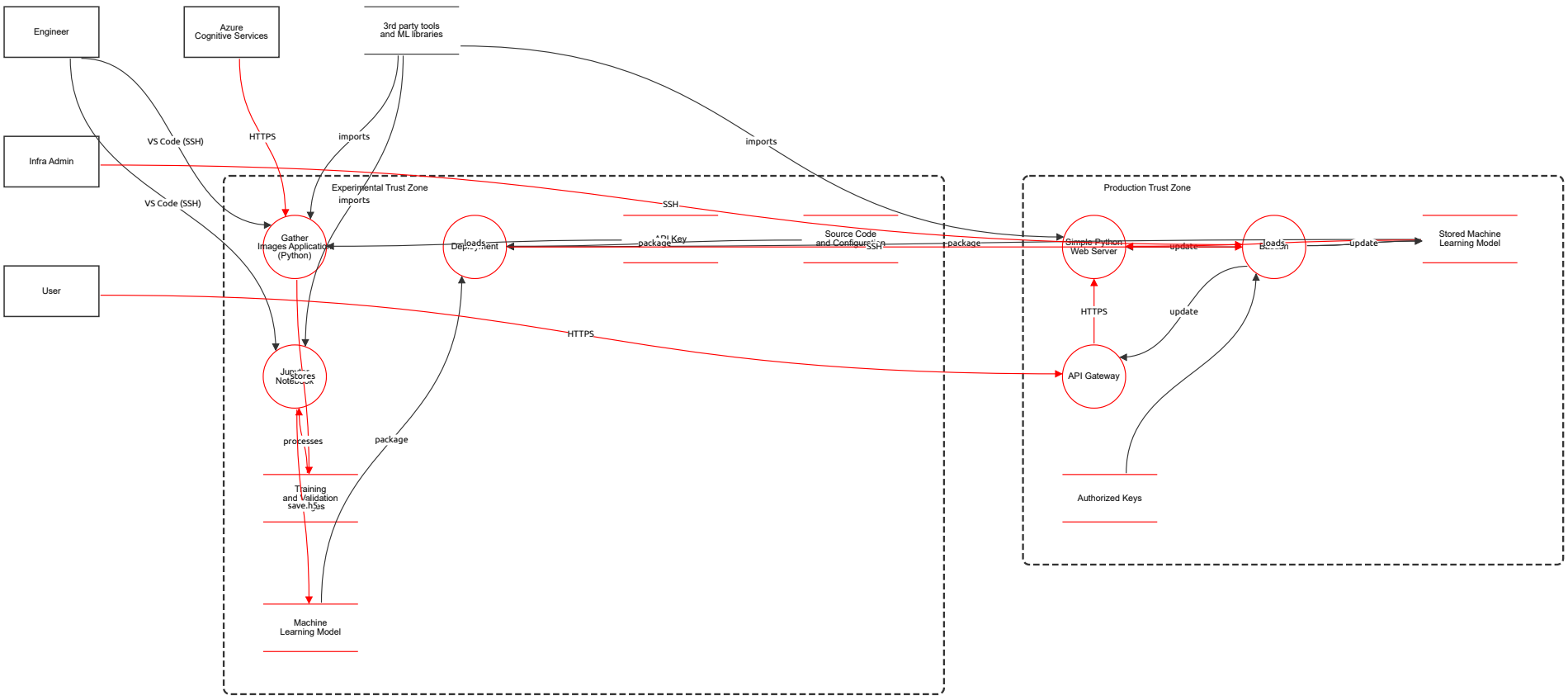
High level system description

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	30
Total Mitigated	0
Total Open	30
Open / Critical Severity	0
Open / High Severity	20
Open / Medium Severity	10
Open / Low Severity	0

Husky AI


Engineer
Azure Cognitive Services
3rd party tools and ML libraries
Infra Admin
User
VS Code (SSH)
VS Code (SSH)
HTTPS
imports
imports
imports
Experimental Trust Zone
Gather Images Application (Python)
Deployment
SSH
API Key
package
Source Code and Configuration
SSH
package
Production Trust Zone
Simple Python Web Server
update
update
Stored Machine Learning Model
HTTPS
update
API Gateway
HTTPS
stores
processes
package
Training and validation save.h5as
Machine Learning Model
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
	Compromised Engineer Credentials	Spoofing	High	Open		The Engineer actor connects via SSH to both Gather Images Application and Jupyter Notebook within the Experimental Trust Zone. If the engineer's SSH keys or credentials are compromised, an attacker could impersonate the engineer and gain unauthorized access to critical ML development infrastructure, potentially poisoning training data or stealing proprietary models.	- Enforce multi-factor authentication for SSH access - Use hardware security keys for SSH authentication - Implement session monitoring and anomaly detection - Rotate SSH keys regularly - Use jump hosts with time-limited access tokens

Infra Admin (Actor)

Description: Administrator responsible for securing and maintaining production infrastructure.

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Admin Privilege Abuse	Elevation of Privilege	High	Open		The Infrastructure Admin has SSH access to the Bastion host in the Production Trust Zone, which can update critical production components including API Gateway, Simple Python Web Server, and Stored Machine Learning Model. A malicious or compromised admin could abuse these privileges to manipulate production systems.	- Implement privileged access management (PAM) solution - Enforce just-in-time access with approval workflows - Enable comprehensive audit logging of all admin actions - Implement break-glass procedures with alerts - Use separate admin accounts with MFA

Azure Cognitive Services (Actor)

Description: External service providing resources for machine learning experimentation.

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Supply Chain Attack via External Service	Tampering	Medium	Open		Azure Cognitive Services is an external actor providing images via HTTPS to the Gather Images Application. If the service is compromised or serves malicious content, it could poison the training dataset with adversarial images designed to create backdoors in the ML model.	- Implement content validation and sanitization for all external data - Use cryptographic signatures to verify data integrity - Maintain local cache with versioning for rollback capability - Implement anomaly detection on incoming data - Regular security assessments of third-party dependencies

User (Actor)

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

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Unauthenticated User Access	Spoofing	High	Open		The User actor connects to the API Gateway via HTTPS from outside the Production Trust Zone. Without proper authentication mechanisms visible in the model, attackers could potentially impersonate legitimate users to access the HuskyAI classification service.	- Implement OAuth 2.0 or JWT-based authentication - Deploy API key management with rate limiting - Use mutual TLS for high-security clients - Implement CAPTCHA for public endpoints - Deploy WAF to filter malicious requests

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
	Code Injection in Image Gathering	Tampering	High	Open		The Gather Images Application (Python) receives data from external Azure Cognitive Services and processes it before storing in Training and Validation Images. Malformed or malicious image data could exploit vulnerabilities in image processing libraries, leading to remote code execution within the Experimental Trust Zone.	- Implement input validation and sanitization - Run image processing in sandboxed environment - Keep all dependencies and libraries updated - Use static and dynamic code analysis tools - Implement least privilege principle for process execution
	API Key Exposure	Information Disclosure	High	Open		The Gather Images Application loads API keys from API Key storage to authenticate with Azure Cognitive Services. If the application logs, error messages, or memory dumps expose these keys, attackers could use them to access external services or exhaust API quotas.	- Use secure key management service (KMS) - Implement key rotation policies - Never log sensitive credentials - Use environment variables or secure vaults - Monitor API key usage for anomalies

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
	Notebook Code Execution Vulnerability	Elevation of Privilege	High	Open		Jupyter Notebook in the Experimental Trust Zone processes training images and accepts code from Engineers via SSH. Malicious code in notebooks could execute with elevated privileges, potentially accessing sensitive data across the experimental environment or manipulating ML models.	- Implement notebook execution sandboxing - Use read-only kernels for production notebooks - Enable audit logging for all notebook executions - Implement code review process for notebooks - Use containerized environments with resource limits
	Model Poisoning Attack	Tampering	High	Open		Jupyter Notebook processes Training and Validation Images and saves models to Machine Learning Model storage. An attacker with access could manipulate training data or model parameters to create backdoored models that misclassify specific inputs in production.	- Implement data provenance tracking - Use differential privacy techniques - Perform model validation and testing - Implement anomaly detection in model behavior - Maintain model versioning with rollback capability

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
	Unauthorized Model Deployment	Elevation of Privilege	High	Open		The Deployment service packages models and code from multiple sources and connects via SSH to Bastion for production deployment. Compromise of this service could allow attackers to deploy malicious models or code to production, affecting all users of the system.	- Implement deployment approval workflows - Use cryptographic signing for deployment packages - Enforce separation of duties for deployment - Implement automated security scanning of packages - Use immutable deployment artifacts

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
	Training Data Tampering	Tampering	High	Open		Training and Validation Images storage in the Experimental Trust Zone contains encrypted business-sensitive data with RBAC controls. However, if access controls are misconfigured or bypassed, attackers could modify training data to create poisoned models that exhibit targeted misclassification behaviors.	- Implement immutable storage with versioning - Use cryptographic checksums for data integrity - Enable audit logging for all data access - Implement anomaly detection for data modifications - Regular access control reviews and principle of least privilege

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
	API Key Theft	Information Disclosure	High	Open		API Key storage contains encrypted credentials with RBAC controls, accessed by Gather Images Application. Despite encryption, if the storage is compromised or access controls fail, attackers could steal API keys to impersonate the application when accessing external services or exhaust API quotas causing financial damage.	- Use dedicated key management service (KMS) - Implement envelope encryption - Enable key usage monitoring and alerting - Implement automatic key rotation - Use temporary credentials where possible

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
	Model Intellectual Property Theft	Information Disclosure	Medium	Open		Machine Learning Model storage in the Experimental Trust Zone contains encrypted business-sensitive models with ACL controls. These models represent valuable intellectual property that could be stolen if storage access controls are compromised, potentially allowing competitors to replicate the HuskyAI capabilities.	- Implement model encryption at rest and in transit - Use digital rights management for models - Implement access logging and monitoring - Use model watermarking techniques - Regular security audits of storage permissions

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
	Source Code and Configuration Exposure	Information Disclosure	High	Open		Source Code and Configuration storage contains encrypted business-sensitive deployment files with RBAC controls. If compromised, attackers could gain knowledge of system architecture, find vulnerabilities in code, or extract sensitive configuration including database credentials, API endpoints, and security settings.	- Separate configuration from code - Use secret management solutions - Implement code obfuscation for sensitive logic - Regular security scanning of code repositories - Implement least privilege access controls

Simple Python Web Server (Process)

Description: Serves as simple web server

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Web Server Remote Code Execution	Elevation of Privilege	High	Open		The Simple Python Web Server in the Production Trust Zone receives requests from API Gateway and loads ML models. Python web servers may be vulnerable to deserialization attacks, path traversal, or code injection, potentially allowing attackers to execute arbitrary code in production.	- Implement input validation and sanitization - Use secure deserialization practices - Deploy in containerized environment with minimal privileges - Implement runtime application self-protection (RASP) - Regular security patching and vulnerability scanning
	Model Inference Data Leakage	Information Disclosure	Medium	Open		The Simple Python Web Server loads models from Stored Machine Learning Model and serves inference requests. Through model inversion or membership inference attacks, attackers could potentially extract training data or model parameters from the API responses.	- Implement differential privacy in model responses - Add noise to model outputs - Rate limit API requests per user - Monitor for suspicious query patterns - Implement model extraction detection

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
	API Gateway Bypass	Elevation of Privilege	High	Open		The API Gateway in the Production Trust Zone is the entry point for external Users and receives updates from Bastion. If the gateway's access controls are misconfigured or bypassed, attackers could directly access internal services like the Simple Python Web Server, circumventing security controls.	- Implement strict API gateway policies - Use Web Application Firewall (WAF) - Enforce mutual TLS for internal communications - Implement API versioning and deprecation - Regular security configuration reviews

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	DDoS Attack on API Gateway	Denial of Service	Medium	Open		The API Gateway receives HTTPS traffic from external Users. Without proper rate limiting and DDoS protection, attackers could overwhelm the gateway with requests, making the HuskyAI service unavailable to legitimate users.	- Implement rate limiting per IP and user - Deploy DDoS protection service - Use CDN for static content - Implement circuit breakers - Auto-scaling based on load metrics

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
	Bastion Host Compromise	Elevation of Privilege	High	Open		The Bastion host in the Production Trust Zone is a critical security component that can update API Gateway, Simple Python Web Server, and Stored Machine Learning Model. If compromised, it provides attackers with broad access to production systems, potentially allowing complete system takeover.	- Harden bastion host with minimal attack surface - Implement host-based intrusion detection - Use ephemeral bastion instances - Enable comprehensive logging and monitoring - Implement network segmentation and micro-segmentation
	SSH Key Management Weakness	Spoofing	High	Open		Bastion loads Authorized Keys from storage for SSH authentication. If these keys are compromised or improperly managed, attackers could gain unauthorized access to the bastion host and subsequently to all production systems it can access.	- Use SSH certificate authority instead of static keys - Implement key rotation policies - Use hardware security modules for key storage - Monitor for unauthorized key usage - Implement time-based access controls

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
	SSH Key Compromise	Information Disclosure	High	Open		Authorized Keys storage in the Production Trust Zone contains encrypted SSH credentials with RBAC controls used by Bastion. Compromise of these keys would allow attackers to impersonate authorized administrators and gain unrestricted access to production infrastructure.	- Use SSH certificate authority - Implement hardware security modules - Enable key usage auditing - Implement emergency key revocation procedures - Use time-limited certificates

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
	Production Model Tampering	Tampering	High	Open		Stored Machine Learning Model in the Production Trust Zone contains unencrypted business-sensitive models with RBAC controls. The lack of encryption and multiple update paths (from Bastion) create risk of model tampering that could affect all production inference requests.	- Enable encryption at rest immediately - Implement model signing and verification - Use immutable storage with versioning - Implement model integrity checks before loading - Monitor for unauthorized model changes

HTTPS (Data Flow)

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

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Man-in-the-Middle Attack on Image Transfer	Tampering	Medium	Open		The HTTPS flow from Azure Cognitive Services to Gather Images Application crosses from external to the Experimental Trust Zone. While encrypted, without certificate pinning, attackers could perform MITM attacks to inject malicious images into the training pipeline.	- Implement certificate pinning - Use mutual TLS authentication - Validate SSL/TLS certificates - Monitor for certificate changes - Implement content integrity verification

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
	Unencrypted Internal Data Transfer	Information Disclosure	Medium	Open		The flow from Gather Images Application to Training and Validation Images storage is unencrypted within the Experimental Trust Zone. An attacker with network access could intercept image data being stored, potentially revealing sensitive training data or allowing data manipulation.	- Implement TLS for all internal communications - Use IPSec or VPN tunnels - Implement network segmentation - Monitor for suspicious network activity - Use encrypted storage protocols

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
	Data Exfiltration During Processing	Information Disclosure	Medium	Open		The unencrypted flow from Training and Validation Images to Jupyter Notebook within the Experimental Trust Zone could allow attackers with network access to intercept sensitive training data during processing, potentially revealing proprietary datasets.	- Enable encryption for all data transfers - Implement data loss prevention (DLP) controls - Monitor for unusual data access patterns - Use secure enclaves for sensitive processing - Implement network traffic analysis

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
	Model Exfiltration During Save	Information Disclosure	Medium	Open		The unencrypted save.h5 flow from Jupyter Notebook to Machine Learning Model storage within the Experimental Trust Zone could expose proprietary model architectures and weights to attackers with network access during the save operation.	- Implement end-to-end encryption for model transfers - Use secure model serialization formats - Implement network segmentation - Monitor for unauthorized model access - Use encrypted storage protocols

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
	API Request Injection	Tampering	Medium	Open		The HTTPS flow from external User to API Gateway crosses into the Production Trust Zone. Without proper input validation, attackers could inject malicious payloads in API requests to exploit vulnerabilities in downstream services.	- Implement comprehensive input validation - Deploy Web Application Firewall (WAF) - Use API schema validation - Implement request sanitization - Monitor for injection attack patterns

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
	Internal API Exploitation	Tampering	Medium	Open		The unencrypted HTTPS flow from API Gateway to Simple Python Web Server within the Production Trust Zone could be intercepted or manipulated by an attacker with internal network access, potentially modifying API requests or responses.	- Implement mutual TLS between internal services - Use service mesh with automatic encryption - Implement request signing and verification - Monitor for suspicious internal traffic - Use zero-trust network architecture

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 Loading Manipulation	Tampering	High	Open		The unencrypted flow from Stored Machine Learning Model to Simple Python Web Server in the Production Trust Zone could allow attackers with internal access to intercept or modify models during loading, potentially serving corrupted models to production users.	- Implement model encryption in transit - Use cryptographic signatures for model verification - Implement secure model loading protocols - Monitor model loading operations - Use integrity checks before model deployment

SSH (Data Flow)

Description: Transfer sensitive data from Deployment Service to Bastion

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Deployment Channel Compromise	Elevation of Privilege	High	Open		The SSH flow from Deployment Service in Experimental Trust Zone to Bastion in Production Trust Zone crosses trust boundaries. This critical deployment path could be exploited to push malicious code or models from development to production if SSH authentication is compromised.	- Implement deployment signing and verification - Use separate deployment credentials with MFA - Implement approval workflows for production deployments - Monitor all cross-boundary communications - Use air-gapped deployment processes for critical updates

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
	Administrative Access Abuse	Elevation of Privilege	High	Open		The SSH flow from Infrastructure Admin to Bastion crosses into the Production Trust Zone. This privileged access path could be abused by malicious insiders or compromised admin accounts to gain unauthorized control over production systems.	- Implement privileged access management (PAM) - Use time-limited access tokens - Enable session recording and monitoring - Implement break-glass procedures - Use multi-person authorization for critical operations

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