Infosecotb.com with vMeNext Threat Model

Owner: InfoSecOTB
Reviewer: Piotr Kowalczyk
Contributors:
Date Generated: Mon Oct 06 2025

Executive Summary

High level system description

Infosecotb.com is a professional cybersecurity blog hosted on WordPress through BlueHost. The blog serves as a platform for sharing insights, articles, and resources related to information security, targeting cybersecurity professionals and enthusiasts.

Website Structure:
- Content Management System (CMS): Built on WordPress, allowing for easy content creation, management, and publishing.
- User Interaction: Features such as chatbot, comments, contact forms, and newsletter subscriptions that facilitate user engagement.
- Categorized Content: Articles are organized into categories based on topics

Functionality:
- Article Publishing: Regularly updated with new blog posts that include technical guides, best practices, and industry insights.
- Search Functionality: Allows users to search for specific topics or articles.
- Social Media Integration: Links to social media platforms for sharing and promoting content.
- vMeNext AI powered chatbot

User Types:
- Visitors: General users seeking information on cybersecurity topics.
- Administrators: Individuals with backend access for managing content, settings, and website security.

Technical Environment:
- Hosting: Utilizes BlueHost for hosting, which provides shared or dedicated server resources.
- Plugins and Themes: Employs various WordPress plugins for enhanced functionality (e.g., SEO, analytics, security).
- Database: Relies on a MySQL database for storing content, user information, and site settings.
- vMeNext chatbot published using iFrames

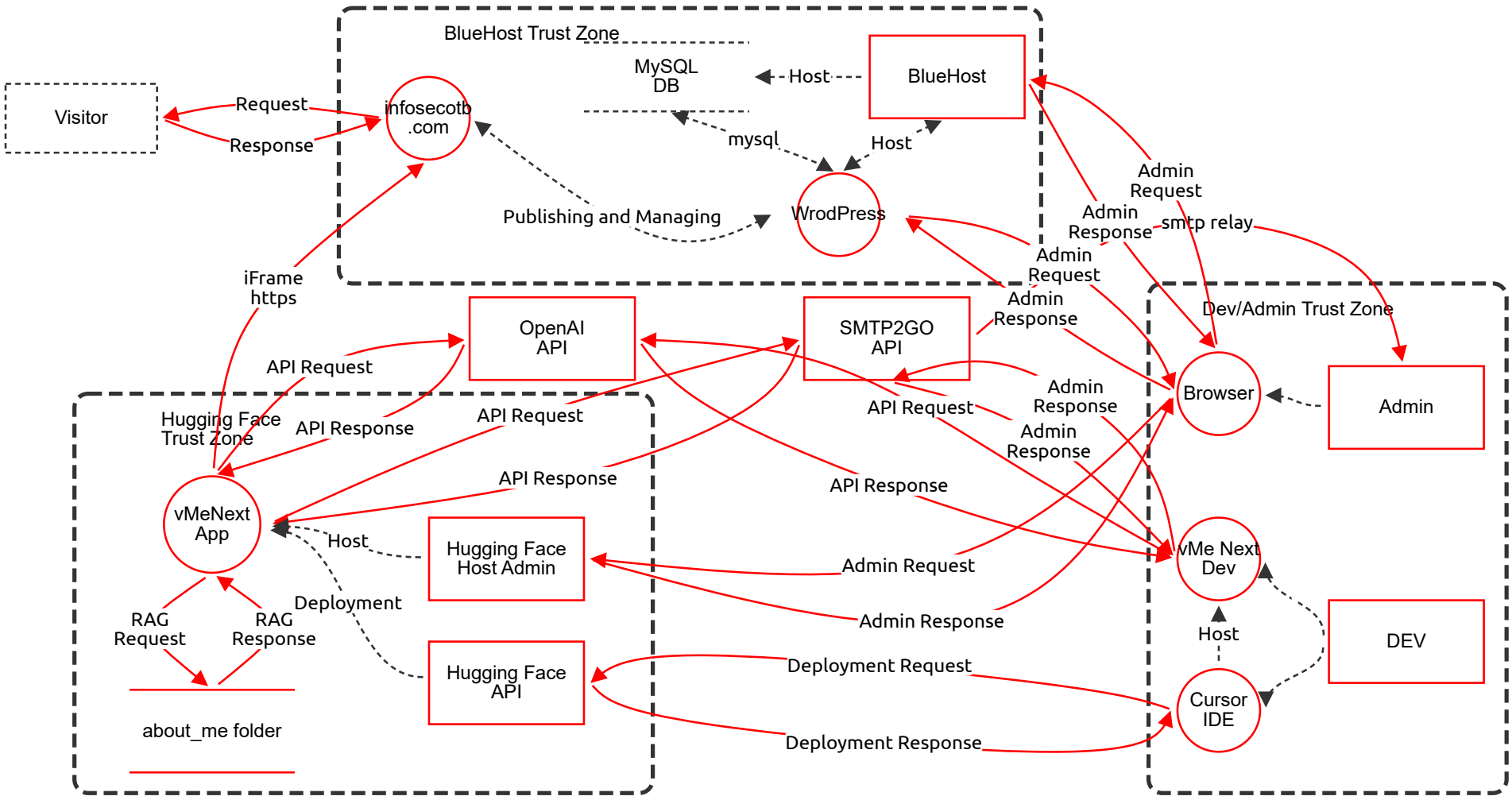
vMeNext is a comprehensive AI-powered chatbot system designed to serve as an intelligent interface for blog content and website management. Built with modern Python technologies, it combines the power of OpenAI's GPT models with automated web scraping, monitoring, and user engagement features.

Key Capabilities:
- Intelligent Conversations: Powered by OpenAI's latest GPT models for natural, context-aware responses
- Blog Content Integration: Automatic scraping, processing, and summarization of blog posts
- Website Monitoring: Continuous availability checking with real-time alerts
- Document Processing: Support for multiple file formats (PDF, DOCX, TXT, MD)
- User Engagement: Automated email notifications and contact management
- Analytics Dashboard: Website uptime statistics with visualizations

Summary

Total Threats	46
Total Mitigated	0
Total Open	46
Open / Critical Severity	0
Open / High Severity	26
Open / Medium Severity	20
Open / Low Severity	0

Infosecotb.com with vMeNext Diagram



Infosecotb.com with vMeNext Diagram

Visitor (Actor) - *Out of Scope*

Reason for out of scope:

Description: Visitor connecting to infosecotb.com using a browser

Number	Title	Type	Severity	Status	Score	Description	Mitigations

vMeNext App (Process)

Description: Gradio ChatBot Python Application with RAG Running on Hugging Face Space

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Unauthorized Access to vMeNext App	Spoofing	High	Open		The vMeNext App process accepts connections via iFrame from the infosecotb.com website. Without proper authentication mechanisms, malicious actors could spoof legitimate requests and gain unauthorized access to the chatbot functionality, potentially extracting sensitive information or manipulating responses.	- Implement OAuth 2.0 or JWT-based authentication for iFrame communications - Validate origin headers and implement CORS policies - Use cryptographic signatures for request validation
	API Key Exposure in Client-Side Code	Information Disclosure	High	Open		The vMeNext App makes API requests to OpenAI API and SMTP2GO API. If API keys are embedded in client-side code or inadequately protected, they could be exposed through browser inspection or network traffic analysis, leading to unauthorized API usage and potential data breaches.	- Store API keys in secure environment variables on the server-side - Implement a backend proxy service to handle API calls - Use key rotation and monitoring for unusual API usage patterns
	Prompt Injection Attacks	Tampering	Medium	Open		The vMeNext App processes user inputs and sends them to OpenAI API. Malicious users could craft inputs that manipulate the AI's behavior, bypass safety measures, or extract information from the RAG context stored in the about_me folder.	- Implement input validation and sanitization - Use prompt engineering techniques to prevent injection - Apply rate limiting and content filtering - Monitor for suspicious patterns in user queries
	Resource Exhaustion via API Abuse	Denial of Service	Medium	Open		The vMeNext App makes multiple external API calls to OpenAI API and SMTP2GO API. Attackers could flood the application with requests, causing excessive API usage, depleting rate limits, or incurring significant costs.	- Implement rate limiting per user/IP address - Use CAPTCHA for suspicious activity - Set up cost alerts and usage caps - Implement request queuing and throttling

about_me folder (Store)

Description: Folder with documents read by Python application and provided to AI ChatBot as a prompt context.

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Unauthorized Access to RAG Documents	Information Disclosure	Medium	Open		The about_me folder store contains documents used for RAG context. If access controls are not properly configured, attackers could potentially access sensitive business information or personal data stored in these documents through directory traversal or path manipulation attacks.	- Implement strict file access permissions - Use principle of least privilege for application access - Encrypt sensitive documents at rest - Implement access logging and monitoring
	Data Poisoning in RAG Context	Tampering	Medium	Open		The about_me folder provides context to the AI chatbot. If an attacker gains write access to this folder, they could modify or inject malicious content that would be used as trusted context, leading to misinformation or harmful responses from the chatbot.	- Implement file integrity monitoring - Use digital signatures for document verification - Restrict write access to authorized administrators only - Maintain versioning and backup of documents

DEV (Actor)

Description: vMeNext Application Developer

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Compromised Developer Account	Elevation of Privilege	High	Open		The DEV actor has privileged access to both Cursor IDE and vMe Next Dev processes within the Dev/Admin Trust Zone. If developer credentials are compromised, attackers could gain full control over the application development and deployment pipeline.	- Enforce multi-factor authentication (MFA) - Implement privileged access management (PAM) - Use hardware security keys for authentication - Regular security awareness training for developers

Cursor IDE (Process)

Description: Cursor IDE used for developing and running vMe Next Dev application and deploying on Hugging Face Space

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Supply Chain Attack via IDE	Tampering	High	Open		The Cursor IDE process is used for developing and deploying the vMeNext application. Malicious extensions, compromised dependencies, or IDE vulnerabilities could inject malicious code into the application before deployment to Hugging Face Space.	- Verify IDE and extension signatures - Use dependency scanning and vulnerability assessment - Implement code signing and verification - Regular security updates for development tools
	Exposed Development Secrets	Information Disclosure	High	Open		The Cursor IDE handles deployment requests to Hugging Face API. Development secrets, API keys, or credentials might be inadvertently committed to version control or exposed through IDE configuration files.	- Use secret scanning in CI/CD pipeline - Implement git hooks to prevent secret commits - Use secure secret management solutions - Regular rotation of credentials and API keys

infosecotb .com (Process)

Description: InfoSec Outside The Box Cybersecurity Blog created and managed with WordPress CMS with vMeNext AI powered chatbot added using iFrame

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	WordPress Core Vulnerabilities	Elevation of Privilege	High	Open		The infosecotb.com process runs on WordPress CMS. Known vulnerabilities in WordPress core, themes, or plugins could be exploited to gain unauthorized administrative access, especially given its exposure to public internet traffic.	- Keep WordPress core, themes, and plugins updated - Implement Web Application Firewall (WAF) - Regular security scanning and penetration testing - Disable unnecessary features and plugins
	XSS via iFrame Integration	Tampering	Medium	Open		The infosecotb.com website embeds vMeNext chatbot using iFrame. Improper sanitization or misconfigured Content Security Policy could allow cross-site scripting attacks, potentially compromising visitor sessions or stealing credentials.	- Implement strict Content Security Policy (CSP) - Use sandbox attribute for iFrame - Validate and sanitize all user inputs - Regular security headers audit
	Brute Force Attack on Admin Panel	Spoofing	Medium	Open		The infosecotb.com WordPress installation accepts admin requests from the Browser process. Without proper rate limiting or account lockout policies, attackers could attempt brute force attacks on the WordPress admin panel.	- Implement login attempt rate limiting - Use CAPTCHA after failed attempts - Enable two-factor authentication - Change default admin URLs - IP whitelisting for admin access

iFrame https (Data Flow)

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Man-in-the-Middle Attack on iFrame Communication	Information Disclosure	Medium	Open		The iFrame https flow between vMeNext App and infosecotb.com crosses trust boundaries. Although using HTTPS, without certificate pinning or additional verification, attackers could potentially intercept or modify the communication.	- Implement certificate pinning - Use Subresource Integrity (SRI) checks - Validate message origin in postMessage handlers - Implement end-to-end encryption for sensitive data

(Data Flow) - *Out of Scope*

Reason for out of scope:

Number	Title	Type	Severity	Status	Score	Description	Mitigations

(Data Flow) - *Out of Scope*

Reason for out of scope:

Number	Title	Type	Severity	Status	Score	Description	Mitigations

(Data Flow) - *Out of Scope*

Reason for out of scope:

Number	Title	Type	Severity	Status	Score	Description	Mitigations

Host (Data Flow) - *Out of Scope*

Reason for out of scope:

Number	Title	Type	Severity	Status	Score	Description	Mitigations

RAG Request (Data Flow)

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	RAG Context Extraction	Information Disclosure	Medium	Open		The RAG Request flow from vMeNext App to about_me folder could be exploited to extract entire document contents through carefully crafted queries, potentially exposing sensitive business information.	- Implement query filtering and validation - Use document access controls - Monitor for data extraction patterns - Implement response size limits

RAG Response (Data Flow)

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	RAG Response Manipulation	Tampering	Medium	Open		The RAG Response flow from about_me folder to vMeNext App could be manipulated if the storage is compromised, leading to incorrect or malicious information being provided to users.	- Implement file integrity monitoring - Use checksums for document verification - Regular content audits - Implement version control for documents

mysql (Data Flow) - *Out of Scope*

Reason for out of scope: Managed by BlueHost

Description: Managed and secured by BlueHost

Number	Title	Type	Severity	Status	Score	Description	Mitigations

smtp relay (Data Flow)

Description: E-mail sent to administrator

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Email Spoofing and Phishing	Spoofing	Medium	Open		The smtp relay flow from SMTP2GO API to Admin crosses public networks. Without proper SPF, DKIM, and DMARC configuration, attackers could spoof emails appearing to come from the system, potentially phishing administrators.	- Configure SPF, DKIM, and DMARC records - Implement email authentication protocols - Use encrypted email transmission - Train administrators on phishing recognition

API Response (Data Flow)

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	API Response Manipulation	Tampering	Medium	Open		The API Response flow from SMTP2GO API to vMeNext App traverses public networks. Even with HTTPS encryption, without response validation, attackers could potentially inject malicious responses if they compromise the connection.	- Implement response signature verification - Use certificate pinning for API connections - Validate response schema and content - Monitor for anomalous API responses

API Request (Data Flow)

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	API Key Leakage in Transit	Information Disclosure	High	Open		The API Request flow from vMeNext App to SMTP2GO API carries authentication credentials over public networks. If API keys are transmitted in headers or URLs, they could be logged or intercepted despite HTTPS encryption.	- Use OAuth 2.0 instead of API keys - Implement request signing with HMAC - Rotate API keys regularly - Monitor API usage for anomalies

Admin Response (Data Flow)

Description: SMTP2GO Administration

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Unauthorized SMTP2GO Configuration Changes	Elevation of Privilege	Medium	Open		The Admin Response flow from vMe Next Dev to SMTP2GO API allows administrative control. Compromised developer environments could lead to unauthorized changes in email relay configuration, potentially redirecting emails or exposing sensitive communications.	- Implement role-based access control - Use audit logging for all admin actions - Require approval workflow for critical changes - Monitor for unauthorized configuration changes

Admin Response (Data Flow)

Description: SMTP2GO Administration

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Session Hijacking on Admin Interface	Spoofing	High	Open		The Admin Response flow from SMTP2GO API to vMe Next Dev crosses public networks. Session tokens or cookies could be intercepted, allowing attackers to hijack administrative sessions.	- Use secure, httpOnly, and sameSite cookie flags - Implement session timeout and rotation - Bind sessions to IP addresses - Use anti-CSRF tokens

API Request (Data Flow)

Description: OpenAI API Request

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Excessive OpenAI API Usage	Denial of Service	Medium	Open		The API Request flow from vMe Next Dev to OpenAI API could be abused during development/testing, leading to excessive API calls, rate limiting, or unexpected costs.	- Implement development environment rate limits - Use mock APIs for testing - Monitor API usage and set alerts - Separate development and production API keys

API Request (Data Flow)

Description: OpenAI API Request

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Prompt Leakage to OpenAI	Information Disclosure	Medium	Open		The API Request flow from vMeNext App to OpenAI API transmits user prompts and RAG context over public networks. Sensitive business information or personal data could be exposed to OpenAI's infrastructure.	- Implement data classification and filtering - Anonymize sensitive information before API calls - Use on-premise LLM for sensitive data - Review OpenAI's data retention policies

API Response (Data Flow)

Description: OpenAI API Response

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Malicious AI Response Injection	Tampering	Medium	Open		The API Response flow from OpenAI API to vMeNext App could contain malicious or inappropriate content if the AI model is manipulated or if responses are not properly validated before display.	- Implement content filtering and moderation - Validate and sanitize AI responses - Use response templates and constraints - Monitor for harmful content patterns

Deployment (Data Flow) *- Out of Scope*

Reason for out of scope:

Number	Title	Type	Severity	Status	Score	Description	Mitigations

Host (Data Flow) *- Out of Scope*

Reason for out of scope:

Number	Title	Type	Severity	Status	Score	Description	Mitigations

Response (Data Flow)

Description: Response from infosecotb.com website including vMeNext chatbot

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Visitor Session Hijacking	Spoofing	Medium	Open		The Response flow from Visitor to infosecotb.com over public networks using HTTPS could be vulnerable to session hijacking if proper session management is not implemented.	- Implement secure session management - Use HTTPS-only cookies with secure flags - Implement session timeout and regeneration - Monitor for suspicious session activity

Publishing and Managing (Data Flow) *- Out of Scope*

Reason for out of scope: Managed and secured by BlueHost

Number	Title	Type	Severity	Status	Score	Description	Mitigations

Request (Data Flow)

Description: Request to infosecotb.com website including vMeNext chatbot

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	DDoS Attack on Website	Denial of Service	High	Open		The Request flow from infosecotb.com to Visitor represents public internet exposure. The website could be targeted by distributed denial of service attacks, making it unavailable to legitimate users.	- Implement DDoS protection service (CloudFlare, etc.) - Use rate limiting and traffic filtering - Configure auto-scaling for traffic spikes - Implement CAPTCHA for suspicious traffic

Host (Data Flow) *- Out of Scope*

Reason for out of scope:

Description: Managed by BlueHost

Number	Title	Type	Severity	Status	Score	Description	Mitigations

Host (Data Flow) - *Out of Scope*

Reason for out of scope: Managed and secured by BlueHost

Number	Title	Type	Severity	Status	Score	Description	Mitigations

Admin Response (Data Flow)

Description: WordPress Administration

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	WordPress Admin Privilege Escalation	Elevation of Privilege	High	Open		The Admin Response flow from Browser to WordPress crosses public networks and trust boundaries into the BlueHost zone. Vulnerabilities in WordPress authentication could allow privilege escalation attacks.	- Implement principle of least privilege - Use role-based access control - Regular security audits of user permissions - Monitor for unauthorized privilege changes

Admin Request (Data Flow)

Description: WordPress Administration

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	WordPress Admin Credential Theft	Information Disclosure	High	Open		The Admin Request flow from WordPress to Browser transmits authentication credentials over public networks. Despite HTTPS, credentials could be exposed through phishing, keylogging, or man-in-the-middle attacks.	- Enforce strong password policies - Implement two-factor authentication - Use password managers - Regular security training for administrators

Admin Request (Data Flow)

Description: BlueHost Administration

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	BlueHost Control Panel Compromise	Elevation of Privilege	High	Open		The Admin Request flow from Browser to BlueHost provides hosting infrastructure control. Compromised credentials could give attackers full control over the hosting environment, including database and file access.	- Use dedicated admin workstations - Implement IP whitelisting for admin access - Enable all available security features in BlueHost - Regular security audits of hosting configuration

Admin Response (Data Flow)

Description: BlueHost Administration

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Hosting Configuration Tampering	Tampering	High	Open		The Admin Response flow from BlueHost to Browser could be intercepted or manipulated, potentially showing false configuration states or hiding security breaches in the hosting environment.	- Verify configuration changes through multiple channels - Implement configuration change alerts - Use configuration management tools - Regular configuration backups

Admin Response (Data Flow)

Description: Hugging Face Administration

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Hugging Face Account Takeover	Spoofing	High	Open		The Admin Response flow from Hugging Face Host Admin to Browser crosses trust boundaries. Compromised Hugging Face credentials could allow attackers to modify or delete the deployed application.	- Enable MFA on Hugging Face account - Use API tokens with minimal required permissions - Monitor for unusual login activity - Regular credential rotation

Admin Request (Data Flow)

Description: Hugging Face Administration

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Unauthorized Application Deployment	Tampering	High	Open		The Admin Request flow from Browser to Hugging Face Host Admin allows application deployment control. Attackers could deploy malicious versions of the application if they gain access to admin credentials.	- Implement deployment approval workflow - Use code signing for deployments - Monitor for unauthorized deployments - Implement rollback capabilities

Deployment Request (Data Flow)

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	CI/CD Pipeline Injection	Tampering	High	Open		The Deployment Request flow from Cursor IDE to Hugging Face API represents the CI/CD pipeline. Malicious code could be injected during the build and deployment process.	- Implement secure CI/CD practices - Use signed commits and deployments - Scan for vulnerabilities before deployment - Implement deployment environment isolation

Deployment Response (Data Flow)

Description: Hugging Face Space Application Deployment

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Deployment Rollback Attack	Tampering	Medium	Open		The Deployment Response flow from Hugging Face API to Cursor IDE confirms deployment status. Attackers could manipulate responses to hide failed deployments or rollback to vulnerable versions.	- Implement deployment verification checks - Use immutable deployment logs - Monitor application version in production - Implement automated rollback detection

API Response (Data Flow)

Description: OpenAI API Response

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Development Environment Data Leakage	Information Disclosure	Medium	Open		The API Response flow from OpenAI API to vMe Next Dev in the development environment might contain sensitive data or reveal system internals that should not be exposed in development logs or debugging output.	- Implement data masking in development - Use separate API keys for dev/prod - Clear development logs regularly - Implement secure coding practices

MySQL DB (Store) *- Out of Scope*

Reason for out of scope: Managed by BlueHost

Description: MySQL Database used for WordPress website

Number	Title	Type	Severity	Status	Score	Description	Mitigations

Admin (Actor)

Description: System Administrator

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Admin Account Compromise	Elevation of Privilege	High	Open		The Admin actor in the Dev/Admin Trust Zone has extensive privileges across multiple systems. A compromised admin account could lead to complete system takeover.	- Implement privileged access management - Use just-in-time access provisioning - Require MFA for all admin actions - Regular security awareness training

vMe Next Dev (Process)

Description: Gradio ChatBot Python Application Development

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Development Environment Compromise	Tampering	High	Open		The vMe Next Dev process handles sensitive API communications with OpenAI and SMTP2GO. A compromised development environment could leak API keys or inject malicious code into production.	- Isolate development environments - Use containerization for development - Implement code review processes - Regular security scanning of development systems

Browser (Process)

Description: Browser used by System Administrator

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Browser-Based Attack on Admin	Tampering	High	Open		The Browser process used by administrators connects to multiple critical systems (WordPress, BlueHost, Hugging Face). Browser vulnerabilities, malicious extensions, or compromised workstations could expose all admin interfaces.	- Use dedicated admin workstations - Implement browser isolation technology - Regular browser and OS updates - Disable unnecessary browser extensions

OpenAI API (Actor)

Description: Artificial Intelligence API secured with a key

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	OpenAI API Key Compromise	Information Disclosure	High	Open		The OpenAI API actor authenticates requests using API keys. If these keys are compromised, attackers could make unauthorized API calls, access conversation history, or incur significant costs.	- Store API keys in secure vaults - Implement key rotation policies - Monitor API usage for anomalies - Use environment-specific keys
	AI Model Poisoning	Tampering	Medium	Open		The OpenAI API processes requests from both vMeNext App and vMe Next Dev. Carefully crafted prompts could potentially influence the model's behavior or extract training data.	- Implement prompt validation and filtering - Monitor for adversarial inputs - Use model versioning and rollback capabilities - Implement response validation

SMTP2GO API (Actor)

Description: E-mail relay hosted system API secured with key

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Email Relay Abuse	Denial of Service	Medium	Open		The SMTP2GO API could be abused to send spam or phishing emails if API credentials are compromised or if the application doesn't properly validate email recipients and content.	- Implement email rate limiting - Validate email recipients - Monitor for unusual email patterns - Use email templates to restrict content

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	SMTP2GO API Key Exposure	Information Disclosure	High	Open		The SMTP2GO API authenticates using API keys. Exposure of these keys could allow attackers to send unauthorized emails, potentially damaging reputation or conducting phishing campaigns.	- Secure API key storage - Regular key rotation - IP whitelisting for API access - Monitor email sending patterns

Hugging Face Host Admin (Actor)

Description: Hugging Face Hosting Administrator Control Panel

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Hugging Face Infrastructure Attack	Elevation of Privilege	High	Open		The Hugging Face Host Admin provides infrastructure control for the vMeNext application. Compromise of this interface could lead to application takeover, data theft, or service disruption.	- Enable all available security features - Regular security audits - Implement infrastructure as code - Monitor for configuration drift

Hugging Face API (Actor)

Description: Hugging Face Deployment API

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Deployment API Abuse	Tampering	High	Open		The Hugging Face API handles application deployments. Unauthorized access could allow attackers to deploy malicious code, delete applications, or access deployment secrets.	- Use deployment tokens with minimal permissions - Implement deployment signing - Audit all deployment activities - Use deployment environments isolation

BlueHost (Actor)

Description: Administrator access to BlueHost

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	Hosting Provider Compromise	Elevation of Privilege	High	Open		The BlueHost actor provides hosting infrastructure control. A compromise at the hosting provider level could expose the entire WordPress installation, database, and stored files.	- Enable all BlueHost security features - Regular backups to external storage - Monitor for unauthorized access - Implement defense in depth strategies

WrodPress (Process)

Description: WordPress Content Management System

Number	Title	Type	Severity	Status	Score	Description	Mitigations
	WordPress Plugin Vulnerabilities	Elevation of Privilege	High	Open		The WordPress process relies on various plugins for functionality. Vulnerable or malicious plugins could provide backdoor access to the entire website and database.	- Audit and minimize plugin usage - Regular plugin updates - Use reputable plugins only - Implement plugin vulnerability scanning
	SQL Injection Attack	Information Disclosure	High	Open		The WordPress process interacts with MySQL DB. Improper input validation in WordPress or plugins could lead to SQL injection attacks, exposing sensitive data or allowing database manipulation.	- Use parameterized queries - Implement input validation - Regular security scanning - Database activity monitoring