

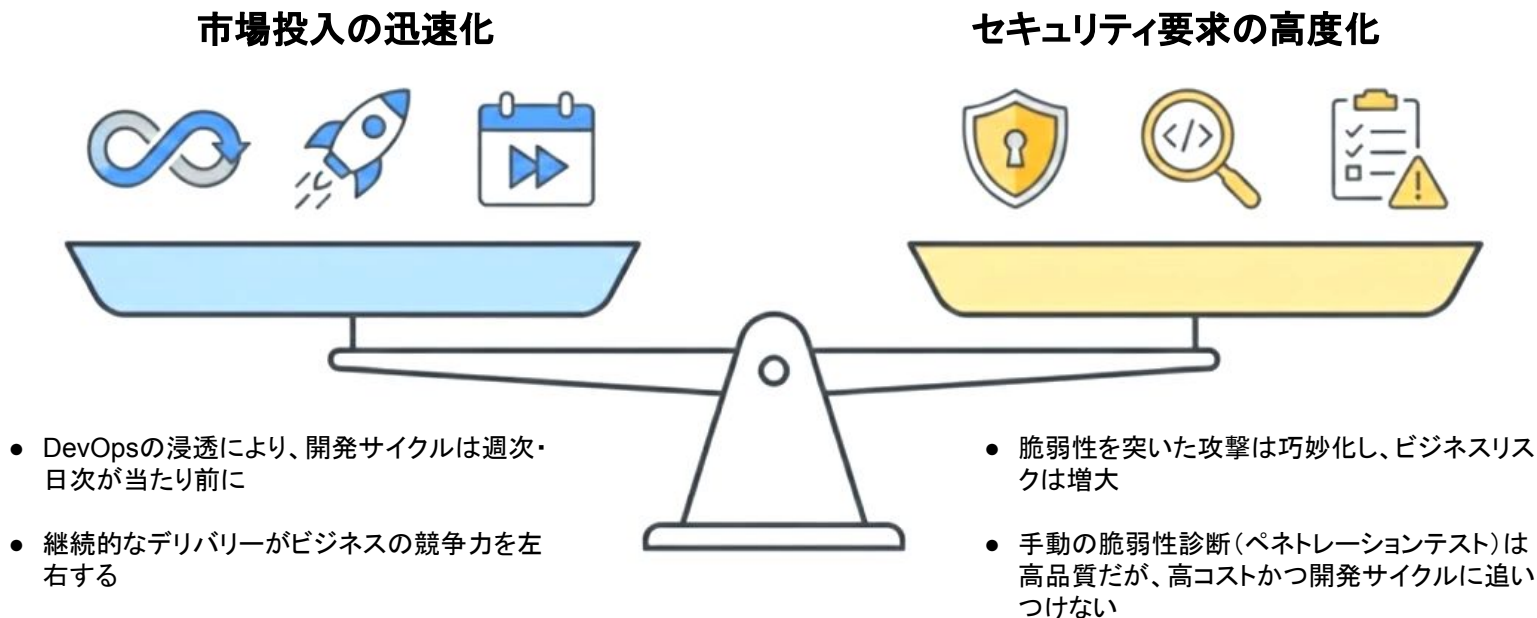
Sun*

Hestia

Security Automation Platform

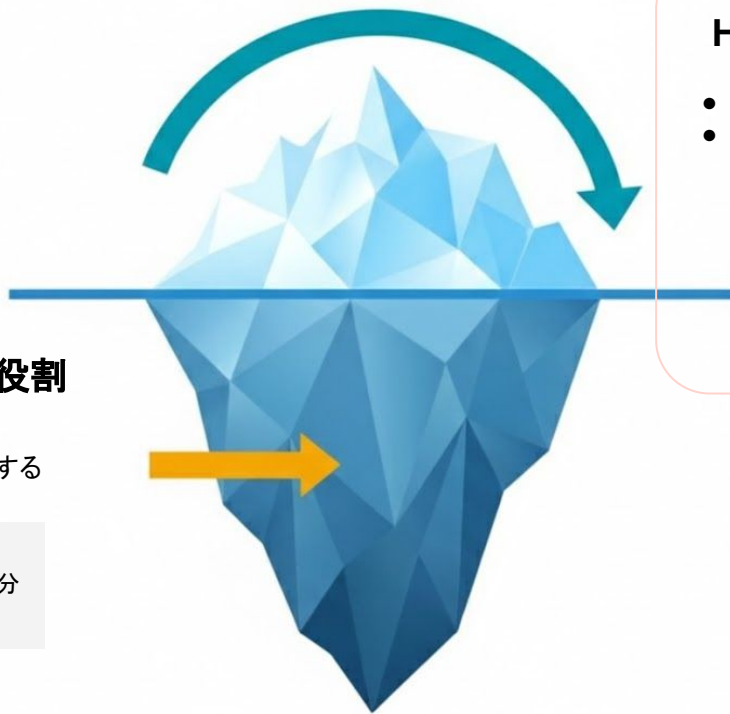
OSSベースの自動脆弱性診断プラットフォーム

現代の開発現場が直面する「速度 vs. 品質」という根本的なジレンマ



開発の加速とセキュリティ要求の増大が、解決困難な緊張関係を生み出している

「自動」で水面上を常時監視し、「手動」はより深く重要な水面下に集中する役割分担が最適解



Hestia(自動診断)の役割

- 広く、浅く、継続的に
- 水面上のリスクを常時監視する
「日常点検」を担う

XSS、SQLi、設定ミス(ヘッダー/CORS)
など、ツールで自動検出可能な表層的な
脆弱性の洗い出しを行う

手動ペネトレーションテストの役割

- 深く、専門的に、定期的に
- 専門家が水面下の重大な脅威に集中する

ビジネスロジックの悪用、認可バイパス
(IDOR/BOLA)、多段フロー攻撃など、高度な分
析に集中すべき「深く専門的な領域」

Hestiaは実績あるOSSスキャナを組み合わせたハイブリッド自動診断を提供

SAST(静的スキャン)

Engine:
Semgrep OSS



- ソースコードを直接解析し、実装レベルの脆弱性(インジェクション、ハードコーディングされた秘密情報など)を検出
- 多数の言語(Python, Java, Go, PHP等)に対応

将来的には Joern, Brakeman, Progpilot等を追加し、カバレッジを拡大予定

DAST(動的スキャン)

Engine: (選択可能)
OWASP ZAP / Burp Dastardly



- 実際にアプリケーションを動作させ、外部からのリクエストを通じて脆弱性(XSS, CSRF, 設定不備など)を検出



これらのスキャン結果を統合

OWASP Top 10の約50~60%を自動検出し、セキュリティの「日常点検」を実現

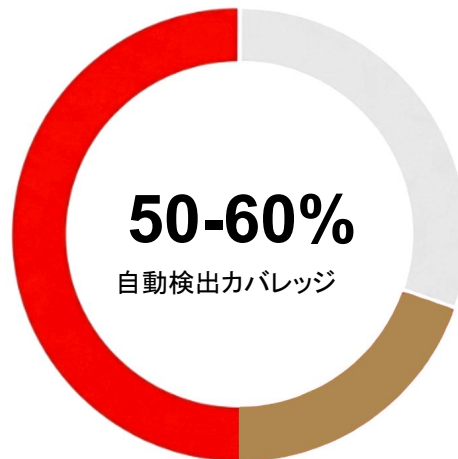
自動検出が強い

- A03: Injection
(インジェクション)
- A05: Security Misconfiguration
(設定ミス)

カバー可能

- A02: Cryptographic Failures
(暗号の不備)
- A10: SSRF

数値は、セキュリティ脅威全体の割合ではなく、OWASP Top 10等の主要な脆弱性項目のうち、ツールによる機械的な判別が可能な「技術的脆弱性」におけるカバー率を指します。



一部は可能だが限界あり

- A01: Broken Access Control
(アクセス制御の欠如)
- A06: Vulnerable Components
(脆弱・古いコンポーネント)
- A08: Software/Data Integrity Failures

自動検出できない(手動診断が必須)

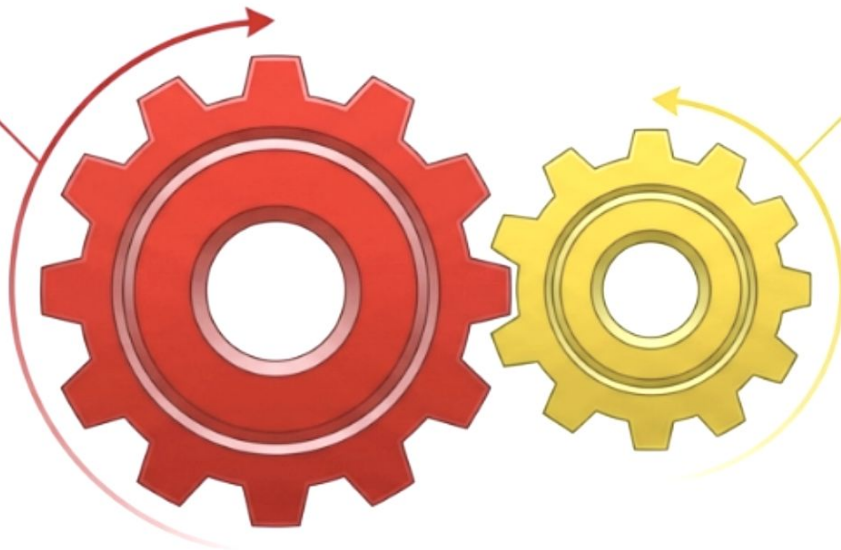
- A04: Insecure Design (安全でない設計)
- A07: Identification/Authentication Failures
(認証の不備)
- A09: Logging & Monitoring Failures
(ログ/監視の不備)

Hestiaは実装ミスや設定ミスといった「技術的な脆弱性」をカバー
設計や複雑な認可ロジックには手動診断が必要

セキュリティ品質は「自動」と「手動」の歯車を組み合わせることで最大化される

Hestia 自動診断

- 網羅的な日常点検
- 毎日/毎週/CI連携
- 対象: 実装ミス・設定ミス・典型脆弱性



手動ペネトレーションテスト

- 攻撃者視点での深掘り評価
- 年1～数回
- 対象: 認可・ビジネスロジック・複雑な攻撃シナリオ

両者を組み合わせることで、診断コストを最適化し、品質を最大化する「DevSecOps」が実現

モバイルアプリ診断の守備範囲

Hestiaでできないこと(モバイル固有領域)



- 端末内部のデータ保存の安全性
- SSL Pinning / 通信盗聴対策
- アプリ改ざん耐性、逆コンパイル耐性
- UI操作を通じた動的解析(DAST)

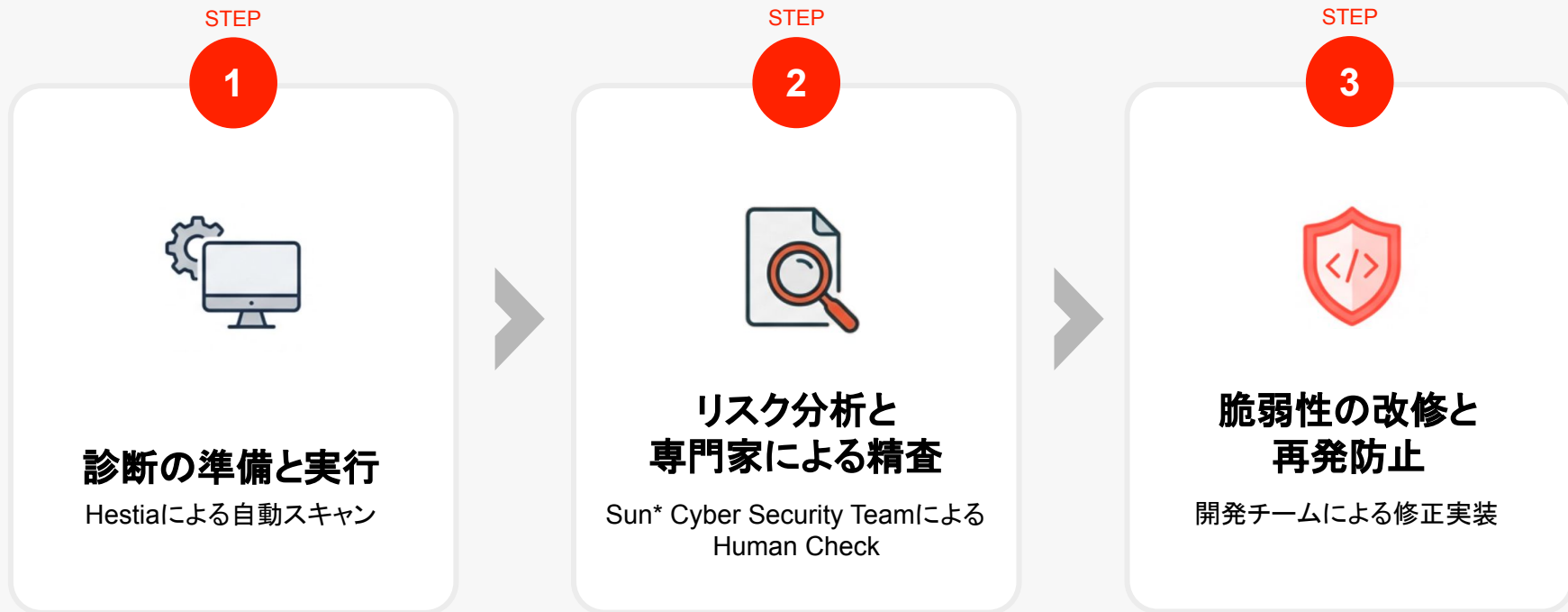
Hestiaでできること(API診断に強み)



- **バックエンド API診断**(◎): モバイルアプリが利用するAPIは、Webアプリと同様のレベルでZAPによる脆弱性診断が可能です。
- **ソースコード解析 (SAST)**(△): SemgrepでKotlin/Swift等のコードを解析し、実装ミスを検出することは設定次第で可能です。

診断実施から改修までの 3 STEP フロー

自動化のスピードと専門家の精査を組み合わせた、効率的なセキュリティ対策サイクル



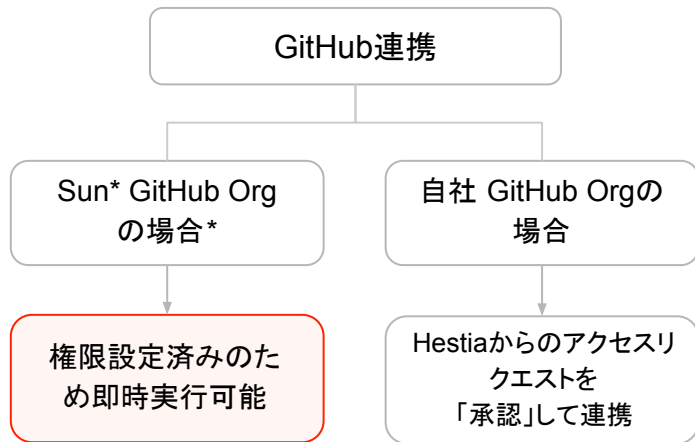
STEP 1 : 診断の準備と実行

システム連携と対象環境の定義



SAST(静的解析)

ソースコード連携



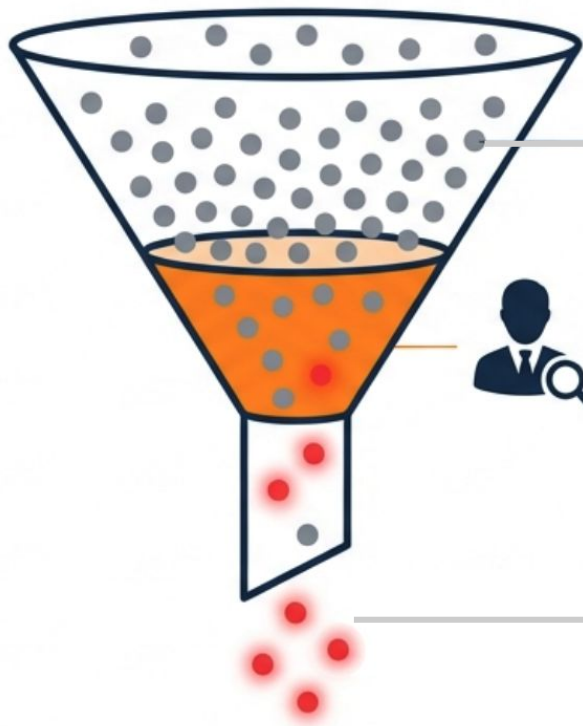
DAST(動的解析)

環境URL連携

対象環境	Staging環境(推奨)のURLを共有
認証情報	Basic認証情報や テスト用ログインアカウントを提供

STEP 2 : リスク分析と専門家による精査

ツール任せにしない「フィルタリング」が、セキュリティ運用の負荷を下げます



Hestiaが脆弱性候補をリストアップ

自動レポート出力：多数の検知項目を抽出

専門家による手動精査

- エンジニアが「これは本当に悪用可能か？」を確認
- False Positive (誤検知) の排除を実行

実行可能な報告書

即時対応が必要な「真正の脆弱性」のみを特定

STEP 3 : 脆弱性の改修と再発防止

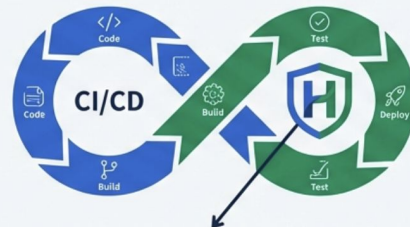
開発チームと連携した迅速な改修と、将来のリスク排除

迅速な改修対応



精査済みレポートに基づき、
Sun*の 開発チームが修正コードを実装
※作業規模により別途お見積りとなる場合があります

再発防止



CI/CDにHestiaを組み込み、
次回のコミットから自動チェック
継続的なセキュリティ品質の担保

⚠ 重要：セキュリティ品質を完結させるための推奨事項

「Hestia(日常点検)」と「手動診断(精密検査)」の役割の違い

項目	Hestia(自動診断) 日常的な健康診断	手動ペネトレーションテスト(推奨) 専門医による精密検査
役割	実装ミスの排除	ロジックの穴を特定
タイミング / 時間	毎日 / 毎週 / 各リリース	重要リリース前 / 年1回の定期診断
限界 / 強み	ビジネスロジック、 複雑な認可フローは対象外	ビジネスロジックの欠陥、 認可制御の不備をカバー

Hestiaですべてのリスクが消えるわけではありません。

高リスクな領域は、専門家による手動診断でカバーすることで、完全なセキュリティ品質が担保されます。

役割分担とプロセスまとめ

Hestia (Tool)



高速なスキャンで
広範囲を網羅

Sun* Security Team



誤検知の排除
リスクの終戦順位付け

Sun* Dev Team



コードの修正、
継続的な監視体制の構築

テクノロジーの速さと、人間の判断力を融合させたセキュリティ体制を提供します。

よくある質問と回答

Q	Hestia(ヘスティア)とは何ですか？
A	Hestiaは、実績あるOSS(オープンソースソフトウェア)である「ZAP(動的解析)」と「Semgrep(静的解析)」を統合したセキュリティ自動化プラットフォームです。単なるスキャンツールではなく、Sun*のCI/CDとも連携しています。
Q	Hestiaを使えば、専門家による手動診断(ペネトレーションテスト)は不要になりますか？
A	いいえ、不要にはなりません。Hestiaは「実装ミス」や「定型的な脆弱性」の検出に優れていますが、人間が業務フローを解釈して行う攻撃(ビジネスロジック攻撃や高度な認可バイパスなど)は検出できません。Hestiaで日常的な「浅く広い」リスクを自動で排除し、年1回などの頻度で専門家による「深く狭い」手動診断を行う ハイブリッド運用 を推奨しています。

Q	どのような脆弱性を検出できますか？
A	OWASP Top 10の約50～60%に相当する「 技術的な脆弱性 」を主に検出します。具体的には、SQLインジェクション、XSS(クロスサイトスクリプティング)、セキュリティヘッダーの不備、CORS設定ミス、ハードコーディングされた秘密情報などが対象です。
Q	モバイルアプリ(iOS/Android)の診断はできますか？
A	限定的な対応となります。HestiaはWeb技術(HTTP/Source Code)を対象としているため、モバイルアプリの「 バックエンドAPI 」や「 ソースコード(Semgrep対応言語) 」の診断は可能です。ただし、 アプリの UI操作、端末内のデータ保存状況、バイナリ解析(改ざん耐性など) といったモバイル固有の領域は診断できません。これらは専用ツールや手動診断が必要です。

よくある質問と回答

Q	認証(ログイン)が必要なページも診断できますか？
A	はい、可能です。Hestiaは認証機能に対応しており、ログイン後の画面もスキャン対象にできます。ただし、多要素認証(MFA)や複雑なログインフローを持つ場合、設定が難しかったり動作が安定しない場合があります。その場合は手動診断での補完を推奨します。
Q	誤検知(False Positive)はありますか？
A	はい、自動診断ツールの特性上、誤検知は発生します。 ツールが報告するIssueは「リスクの可能性がある箇所」であり、必ずしもすべてが悪用可能な脆弱性とは限りません。レポート内容を確認し、精査することを推奨しています。

Q	本番環境でスキャンを実行しても問題ありませんか？
A	ステージング環境での実施を強く推奨します。 DAST(動的解析)のフルスキャンを実行する場合、フォームへのデータ入力や削除といった操作が含まれるため、本番データの整合性に影響を与える可能性があります。 また、大量のリクエストを送信するため、サーバー負荷の観点からも本番環境への実施は慎重に行う必要があります。
Q	導入にあたって、ソースコードへのアクセス権限は必要ですか？
A	解析手法によって異なります。 ・SAST(静的解析) : ソースコードを解析するため、リポジトリへのアクセス権限(Read権限)が必要です。 ・DAST(動的解析) : 稼働しているWebアプリにアクセスするため、URLがあれば実施可能であり、リポジトリ権限は不要です。



Build awesome things, *securely.*