SecHack365

Matchlock

開発駆動コース 仲山ゼミ 齋藤 徳秀

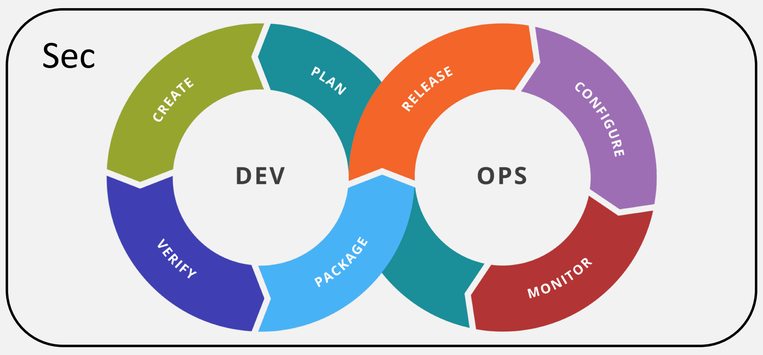
Matchlockとは?

セキュアなWebの開発体制に組み込みやすい
DAST方式のWebセキュリティ検査ツールとその拡張パッケージ群

製作背景と課題

製作背景

多くのWeb系企業で採用されているソフトウェア開発手法であるDevOpsにセキュリティを組み込んだDevSecOpsをはじめとするセキュアな開発体制が近年注目を集めており、多くの場面でセキュアな開発をサポートするツールが運用されています。



製作背景における課題

その取り組みの中でアプリケーションの動的テストでは、IAST方式とDAST方式のツールが利用されており、開発者のセキュアコーディングを助ける役目を担っている。

その中で、IAST方式は対応する言語がまだ少なく、すべてのサービスに対応ができない。そのためDevSecOpsでは、まだDASTの対応できる場面が多いと考えられる。

	IAST Interactive Application Security Testing	DAST Dynamic Application Security Testing
テスト方式	ホワイトボックス	ブラックボックス
属人化の排除	優	劣
検査精度	優	劣
検査時間	短	長
言語依存	有	無
導入コスト	高	低

しかし、既存のDASTツールでは診断員向けの機能が多く、前提の知識が必要であることから属人的になってしまう。また、必要に応じた拡張が行いにくく、ベンダーが対応やツールへの実装が行われるまで待たなければならない。そういった課題を解決をしたいと思いこの開発を始めた。

なぜIASTではなくDASTなの？


検査にIASTは言語に依存する
DASTは言語に依存しない

DASTは導入時に
サービスなどに手を加えない

既存DASTの課題

前提の知識と
訓練の必要性から
属人的

一部のツールを除く
外部連携を含む
拡張性が乏しい

モダンなWeb環境への
対応が進んでいない

さらに開発を進めるにつれ、このDASTの利用されるセキュアな開発体制においても課題があり、それらを解決しなければ、検査の組み込みが行えないのではと思う箇所があり、これらの解決に向けても同時に進めていきました。

既存のDevSecOps テスト段階における課題

テストケースの収集に
時間がかかる

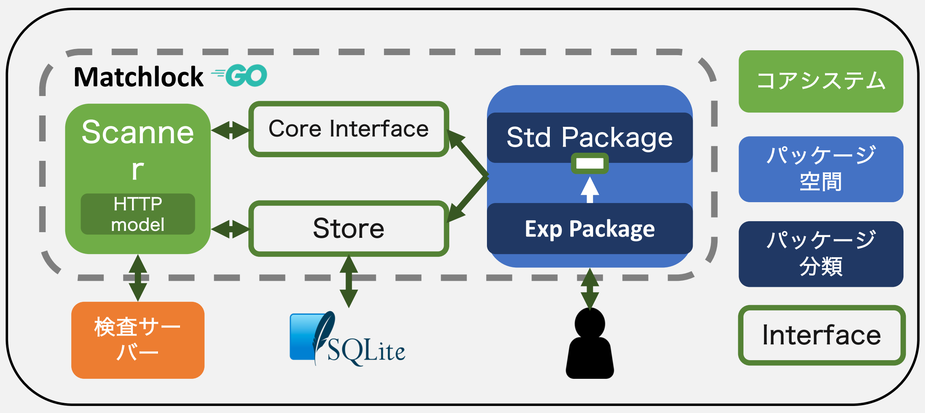
テスト段階に
人の手が介在する

製作背景における課題へのアプローチ

製作背景における課題へのアプローチとして、次のようなアーキテクチャを取ることで解決をはかりました。

まず初めに外部連携を含む拡張性が乏しいという点では、パッケージ方式をアーキテクチャとして採用することにより解決ができるとお考えております。また、モダンなWeb環境への対応が進んでいないという部分でも、このパッケージ方式を用いたヘッダレスブラウザの組み込みで対応することができます。

次に、前提知識と訓練が必要で属人的であるという点では、検査に必要な機能を絞り込むことにより、学習コストを減らし属人的な操作を減らすことができます。

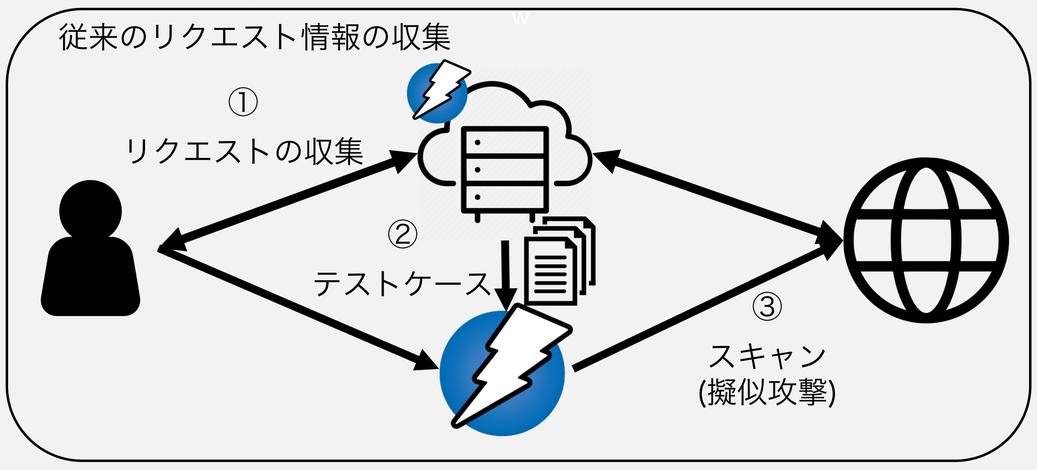


提案手法

Document Base DAST

従来のリクエスト情報の収集方法では①のようにProxyを経由し、Webサイトの巡回やクローラーを用いてリクエスト情報の収集を行い、リクエスト情報をもとに②のテストケースを生成します。その後利用者は操作をおこない、検査のために④スキャン(擬似攻撃)を行いセキュリティテストを行います。

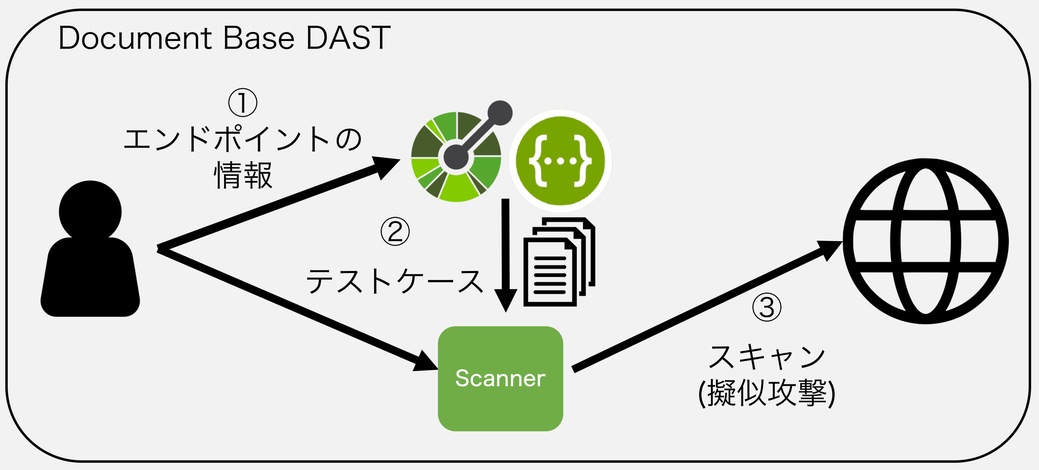
しかし、この収集方法ではリクエストの収集に時間がかかってしまい、さらに人の手が介在してしまうことで、DevSecOpsのような開発体制には組み込みにくくなってしまいます。



これら課題に対して、表題の通りにDocumentを用いてこのDASTにおけるテストケースの生成を行い解決ができるのではないかと思い、次のような手法を提案しました。

①OpenAPIをはじめとしたエンドポイントの仕様書を利用し、②テストケースを生成し、スキャナーはそれを利用しテスト対象のサイトに③スキャン(擬似攻撃)を行いセキュリティテストを行います。

私は、この手法を従来のDASTと区別をつけるため、Document Base DASTと名付けました。



また、このドキュメントベースを用いることで、CIにも組み込むことが容易になり、今まで組み込み難かったDevSecOpsへの組み込みが可能になりました。

NICT 国立研究開発法人 情報通信研究機構