企業に最適なWebシステム構成

ゼンド・ジャパン株式会社

ハードウエアとOS

LAMP–LAPP

WIMP–WISP

SAMP

Windows
Linux
Solaris x86

Intel / AMD

Power

iAMP

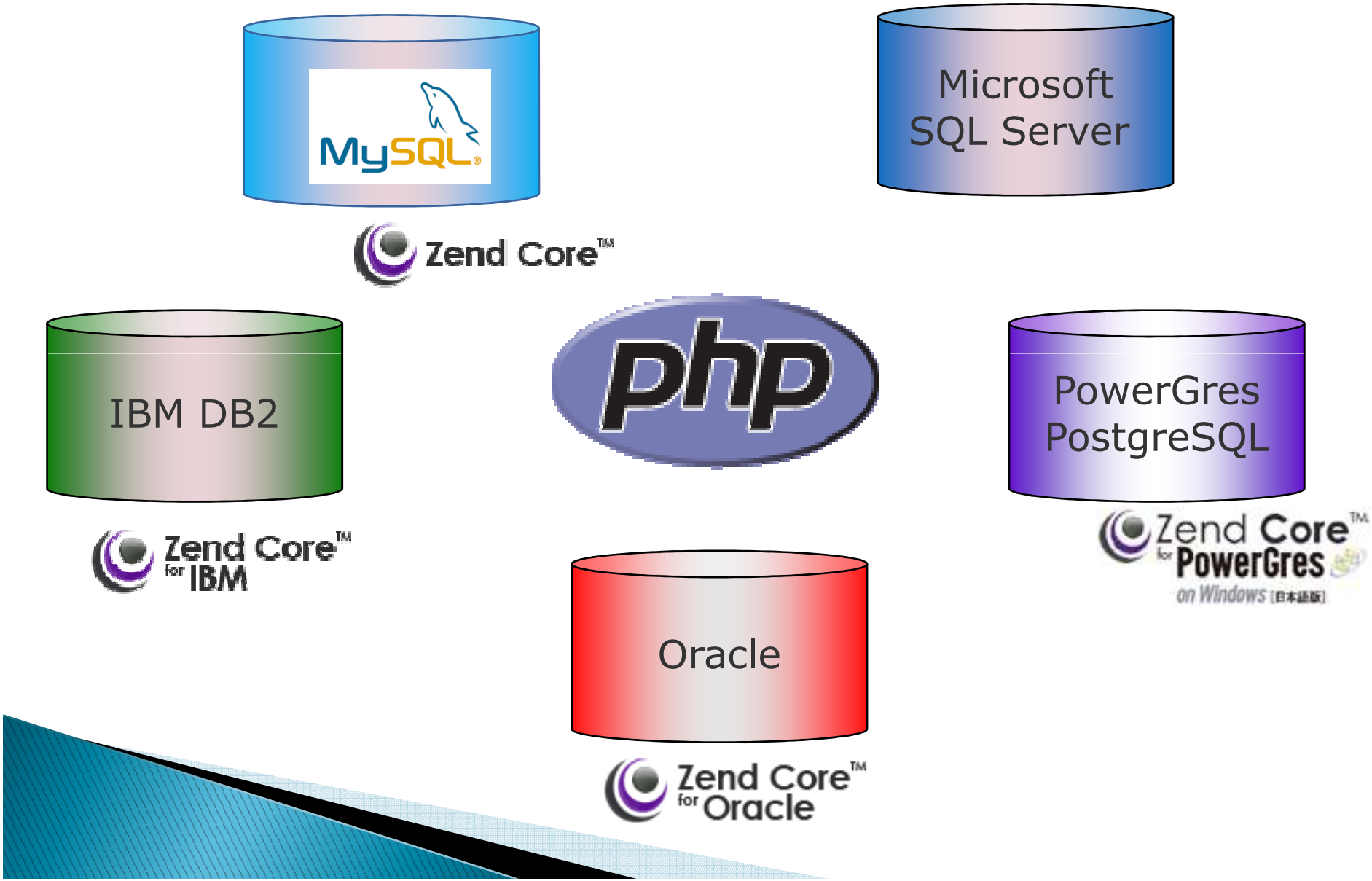
Sun SPARC

SAMP

IBM i

Sun Solaris

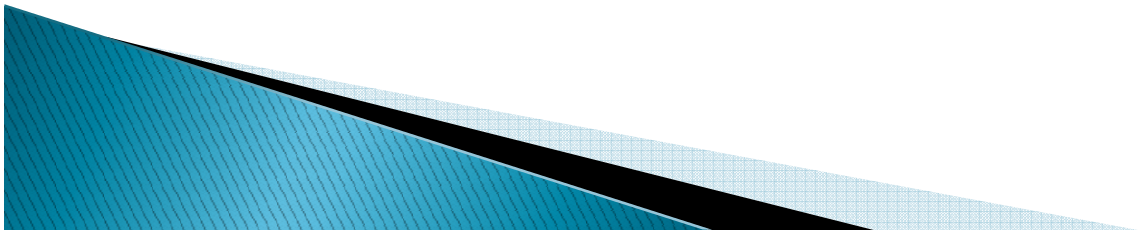
PHPの対応データベース



このセッションの目的

- Webシステムを構築するためのテクノロジーのトレンドを解説します。企業で導入するための実績があるソフトウエア構成をご紹介します。さらにコンプライアンス上の留意点についても言及します。

- LAMPを選択するのが正しいのか
- LAMPがすべてなのか
- その先は何なのか



Webアプリケーションに最適な環境
コンプライアンスのための環境強化

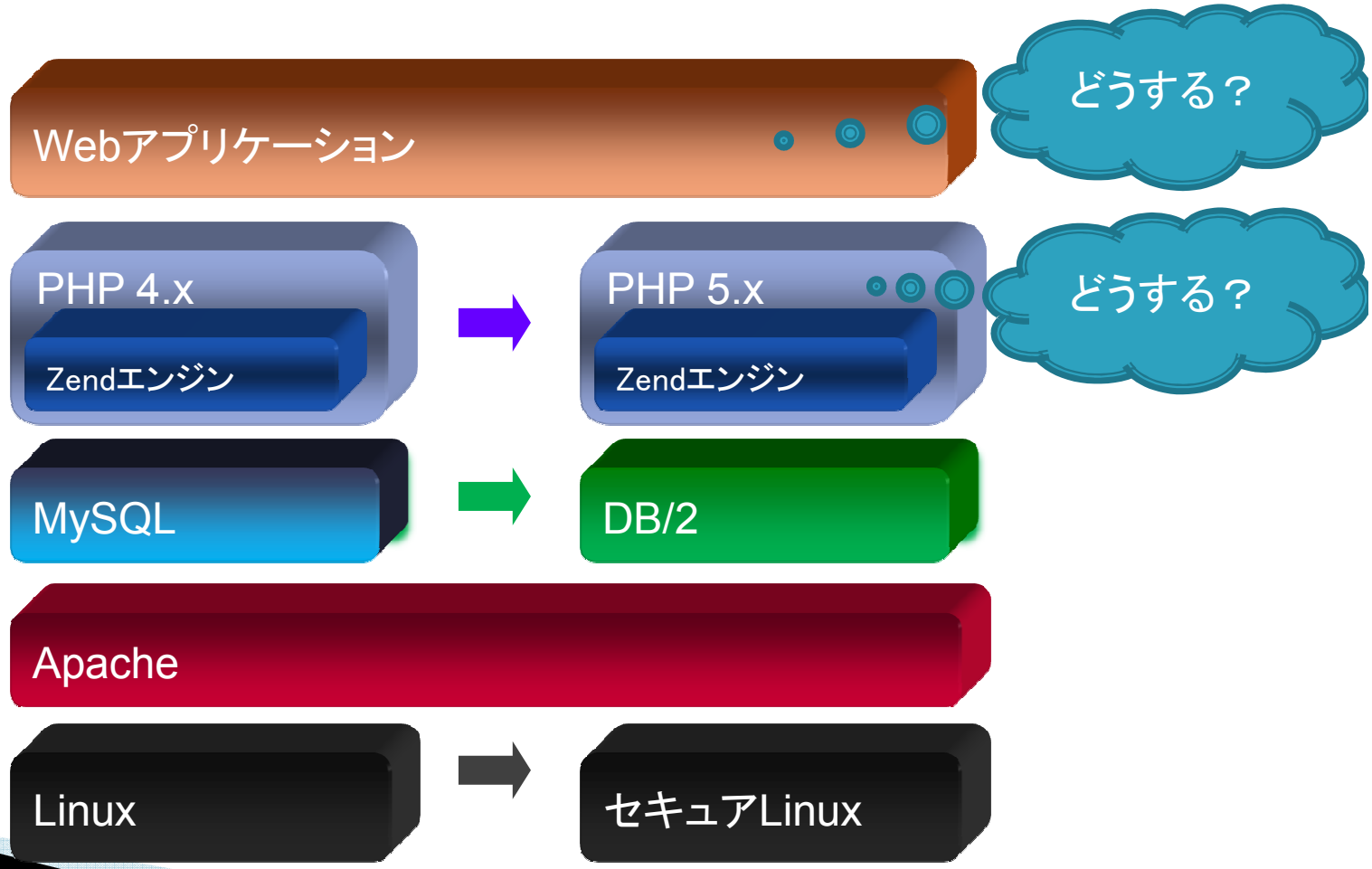
LAMPスタックとは

LAMPスタックとは

- Webアプリケーション環境の代名詞LAMPは、Linux、Apache、MySQL、PHPの組み合わせを表したものです。
- この組み合わせは、実績およびコストで、最も利用されています。

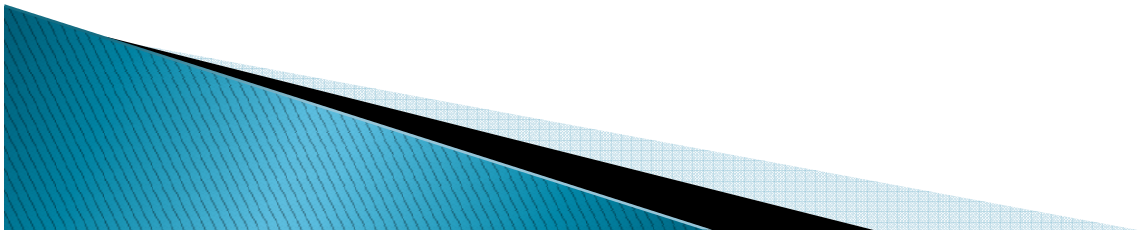
LAMPスタックの強化(補完)

- セキュリティとコンプアライアンスのニーズに応じて置き換えも必要です。



Webシステムのコンサルポイント

	Linux	Apache	MySQL	PHP	アプリケーション
セキュリティ	レ	レ			レ
パフォーマンス	レ	レ	レ		レ
大量アクセス	レ	レ			
コンプライアンス？	レ	レ	レ	レ	レ



LAMPとコンプアライアンス　OSレベル

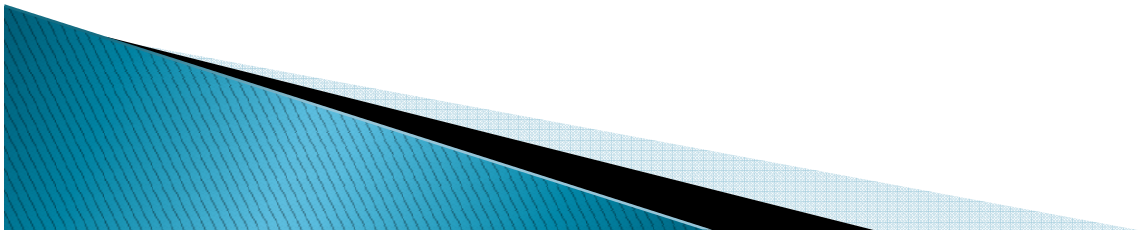
- セキュリティ
  - ネットワーク
  - サービス/プロセス
  - パーミッション/ユーザ
- 運用監視
  - プロセス死活監視
  - 資源管理
  - 侵入検知/改ざん検知
  - アップデート
  - バックアップ監視

設定強化
ユーティリティ導入

LAMPとコンプアライアンス Webサーバ

- ロギング
  - 集計処理
- セキュリティ
  - サーバ証明書
  - アクセス制限

設定強化
ユーティリティ導入



LAMPとコンプアライアンス　データベース

- MySQL 5.1　　→ DB/2 v9.5
  - HA
  - セキュリティ
    - データ暗号化
  - ロギング
    - ログインと処理内容
    - 不正ログイン

ソリューション
交替

LAMPとコンプアライアンス

- PHP 開発環境
  - 統合開発環境　→Zend Studio

- PHPセキュリティ対策
  - コード暗号化　→ Zend Guard

  - コード監査　→ Fortify SCA

- PHP 運用環境
  - 運用監視　→Zend Platform

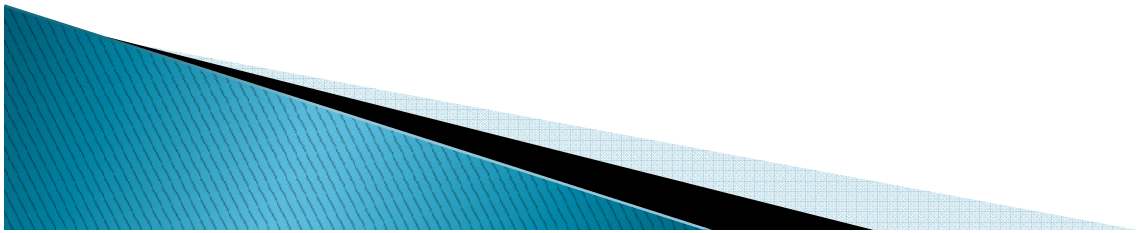
PHPコードレベルの暗号化：
セキュリティとパフォーマンスを向上

Zend Guard

Zend Guard の役割

- PHPのコードは、スクリプト言語なので、テキストファイル形式です。これを暗号化および中間言語化します。

- セキュリティ面の向上
- 実行スピードの向上
- ライセンス機能の付加



Zend Guard 搭載機能

項目	用途	内容
中間コード化	全般	事前に中間コード化を行うことにより、PHPの実行スピードを高速化します。構文チェックも事前に行います。
暗号化	パッケージビジネス セキュリティ向上	PHPソースコードの解析を阻止し、知的財産保護および不正改ざんを防止します。
難読化	パッケージビジネスセキュリティ向上	エラーメッセージ等のシステム情報からソース内部を解析されるのを阻止します。
実行期日設定	パッケージビジネス	実行期日を制限したコードを作成します。 デモ版作成など、営業的な用途に利用されています。
ライセンス機能	パッケージビジネス データセンター	特定のサーバやネットワーク環境に限定して実行できるコードを作成します。

Zend Guard による暗号化

```
<?php @Zend;
4315;
?>
<?php
// test sample
// これはサンプルです
//sample
//PHPヘッダコード
echo("need ZendOptimizer<br>");
?>

<?PHP
/*  !This is not a text file!    */
print <<<EOM
<html><body><a href="http://www.zend.com/store/products/zend-safeguard-suite.php"><img border="0" src=
EOM;
exit();
__halt_compiler();
?>

2006022801 2 0 3 270 608 x
 2  Qゥy)
ay 寒 X再&襦ｼ =ｷ｡｡翁続｡｡ｩ 薬a`dh
f  劒Y  p  ·47ｷR/1X· Z3Cmc 萌 {藾Q 暁抽 HJ · &H! ｪｴ JA  掴K ,9 rhpbnAN齟繹vBu)q ﾌ · · ﾙ
.;^.N褄ﾔﾄ ゆ ﾖ$ｳ ｸ ﾜ`ﾍ W 疼 9ﾃYx 2 ｻ
ｺ s4ﾛ
  ｶ d>ﾌz凵 · J-;洸 · · IA戸 5ﾆp' J; ﾜODs  ﾚ ｵ · HP h2f熙kBﾋ ｵ 更+1 ,4- 溽 iQL[EOF]
```

ヘッダー情報
(Zend Guard が付加)

*処理不能の場合に表示される

コード部分

```
<?php
	$txt = "これはcodeSampleです";
?>

<html>
	<head>
		<title><?php echo($txt);?></title>
	</head>
	<body>
		<?php echo($txt);?>
	</body>
</html>
[EOF]
```

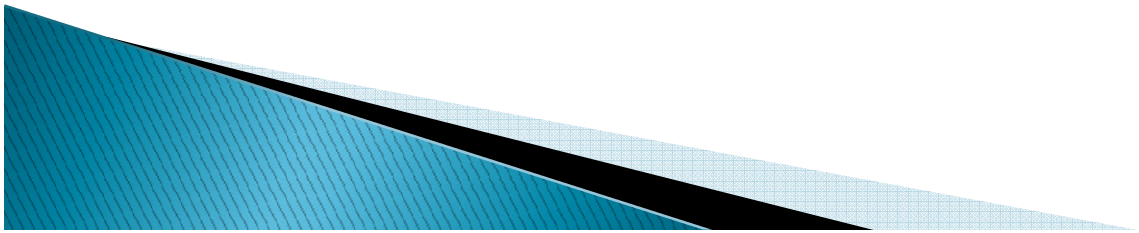
PHPアプリケーションサーバ:
PHPサーバの強化および機能アップ

Zend Platform

Zend Platform の役割

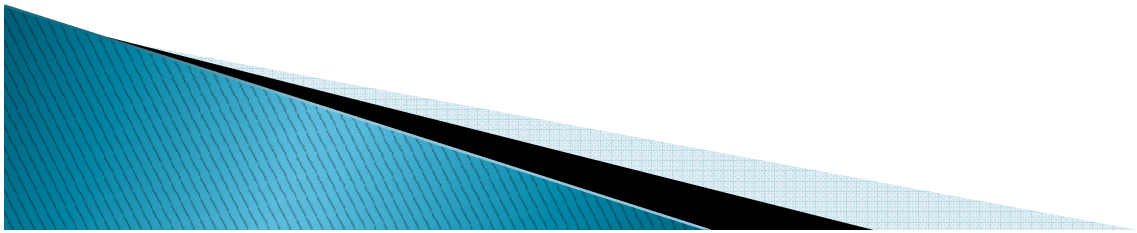
- PHPは、容易に実行できるように簡素な実行環境です。
  運用監視機能など、

- 異常な実行の検出と記録
- キャッシュ機能による実行スピードの向上
- クラスタ構成の実現
- Java実行環境の効率化
- Jobキューの実装



Zend Platform 搭載機能

項目	用途	内容
PHP高速化	全般	PHPの実行スピードを高速化します。 単純設定で、1.5倍から3倍。個別設定で、20倍の高速化を実現します。
PHP実行監視インテリジェンス	全般	PHP標準のエラーログ機能を大幅に拡張し、エラーになる前の予兆の段階から問題を検知します。
Javaブリッジ	中規模以上	PHPからJavaアプリを呼び出す際の効率を飛躍的に向上します。
セッション共有クラスタリング	中規模以上	複数サーバ間でのセッション情報の共有が可能です。
Jobキュー	大規模	PHPプログラムを非同期実行します。時間指定や先行処理指定が可能です。



PHPインテリジェンス トリガー設定

- PHPサーバの様々な事象に対するしきい値を設定し、アラートとして管理を実装しました。サーバの異常挙動をいち早く検出できます。
- PHPサーバ上で発生したエラーは、管理端末にて一括管理できます。エラー発生時の環境を保存することにより、問題解決時に再現することが可能です。
- アラートトリガ
  - スクリプト実行遅延(絶対/相対)
  - PHPエラー、関数エラー
  - 関数実行遅延
  - 余剰メモリ使用(絶対/相対)
  - データベースエラー、クエリ実行遅延
  - アウトプットサイズ不一致
  - 平均負荷過重

PHPインテリジェンス　イベントレポート

- 設定した閾(しきい)値を超えると、詳細なイベントレポートを出力します。イベントレポートには、実行時の環境変数が保存されており、どのような実行状況であったか簡単にわかります。
- Zend Studio と連携して、エラーの再現およびデバックが可能です。

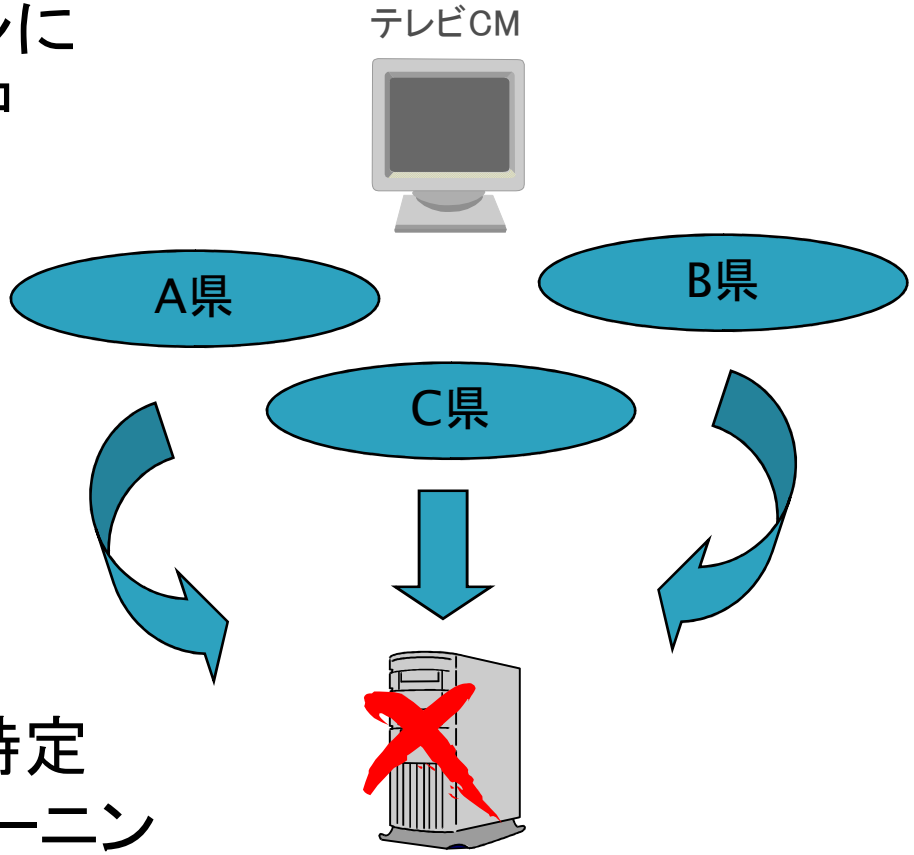
活用事例　株式会社エムティーアイ①

- デコレーションメール用の素材提供サイト『デコとも』
  - 『デコとも』は、文字や背景の色を変更したり、画像を利用してメールをデコレーションする“デコレーションメール”用の素材を1 万点以上提供するサイトでありi モード、EZweb、Yahoo!ケータイ向けに2006 年4 月3 日よりサービスを開始しました。

イメージは、2006年10月時点のものです。

活用事例　株式会社エムティーアイ②

- アクセス集中によるダウン
  - テレビCMなどのプロモーションによって瞬間的なアクセスが集中
  - 動作遅延や応答が発生
  - テレビCMの放映範囲を制限
- Zend Platformに高速化
  - PHP高速化
  - 瞬間的なアクセス集中に対応
  - テレビCMの放映範囲を拡大
- ボトルネックの観測
  - PHP実行遅延でプログラムを特定
  - クエリー実行遅延でDBのチューニング



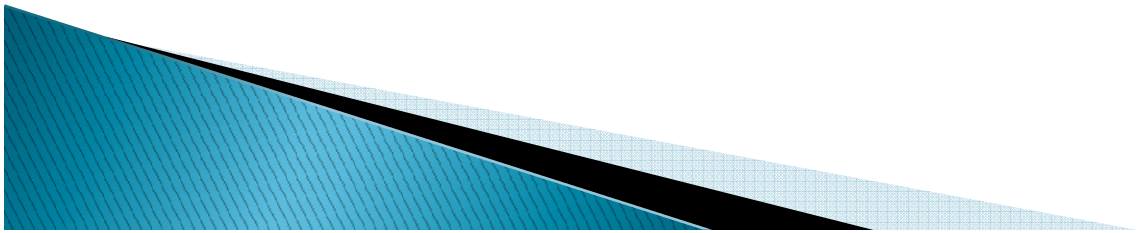
Webアプリケーションの静的セキュリティ監査ツール：
ソースの隅々まで精査する

Fortify SCA

Fortify の役割

- PHPのコードは、フリースタイルでコーディングできます。また、初心者から上級者まで、幅広い開発者がコーディングを行えます。

- 準備のいらない検査スタイル
- 高度な検査ルール(300以上/年4回更新)
- 問題個所をコードレベルで指摘
- 幅広い対応言語



Fortify SCA 構成

Fortify SCA　PHP チェックルール

- Dangerous Function
- PHP Misconfiguration: Missing open_basedir Entry
- PHP Misconfiguration: Missing safe_mode_exec_dir Entry
- PHP Misconfiguration: allow_url_fopen Enabled
- PHP Misconfiguration: allow_url_include Enabled
- PHP Misconfiguration: cgi.force_redirect Disabled
- PHP Misconfiguration: file_uploads Enabled
- PHP Misconfiguration: magic_quotes_gpc Enabled
- PHP Misconfiguration: magic_quotes_runtime Enabled
- PHP Misconfiguration: register_globals Enabled
- PHP Misconfiguration: safe_mode Disabled
- PHP Misconfiguration: session_use_trans_sid Enabled
- Race Condition: PHP Design Flaw
- System Information Leak: PHP Errors
- System Information Leak: PHP Version
- Command Injection
- Cross-Site Scripting

- Dangerous File Inclusion
- Dynamic Code Evaluation: Code Injection
- File Permission Manipulation
- Often Misused: File Uploads
- Path Manipulation
- SQL Injection
- HTTP Response Splitting
- Possible Variable Overwrite: Function Scope
- Possible Variable Overwrite: Global Scope
- Resource Injection
- Setting Manipulation

≪パッケージ関連≫

- DOM
- Error Reporting
- Ereg
- MySQL
- PostgreSQL
- Prereg
- Smarty 2.6.9
- XML

Fortify SCA チェックルール

2008年10月現在で300種類以上

SCAが提供するチェックルールの一部

- SQL Injection
- Cross-Site Scripting
- Buffer Overflows
- Access Control
- Process Control
- No Null Termination
- Setting Manipulation
- Resource Injection
- Password Management
- Unreleased Resource
- Format String Issues
- EJB Resource Permission
- EJB Bad Practices
- J2EE Bad Practices
- Struts Form Field Validation
- Double Memory Free
- Null Pointer Dereference
- Directory Restriction
- Object Model Violation
- Often Misused
- Poor Style
- Access Control
- Insecure Randomness
- Least Privilege Violation
- Code Correctness
- Poor Error Handling
- Dead Code
- J2EE Misconfiguration
- Memory Leak
- Portability Flaw
- Obsolete
- System Information Leak
- Trust Boundary Violation
- ASP.NET Misconfiguration
- Privacy Violations
- Native Callout
- Unsafe Memory Operation
- Unchecked Return Value
- Always Unsafe Functions
- Race Conditions
- Uninitialized Variable
- Session-ID Length
- Entity Bean Configuration
- Information Leakage
- Log Forging
- Integer Overflow

ルールは四半期から半年に一度、ネットワークを通じて最新版を利用することができます

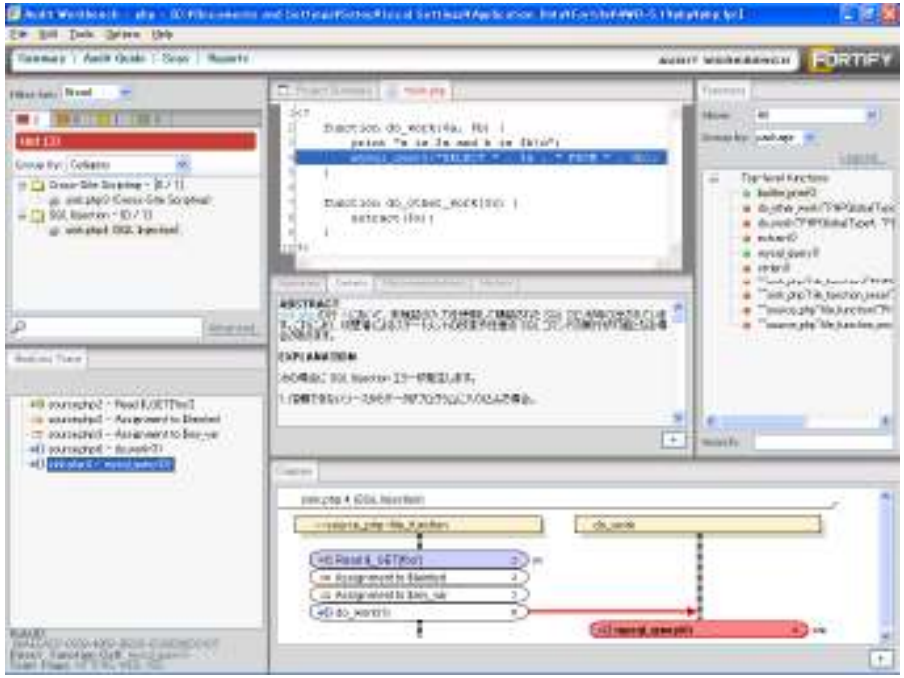
＜拡張ルール＞

.NET	NLOG and Log4Net, Microsoft ASP.NET AJAX (Atlas)
C/C++	MFC™ and ATL™ Glib, Microsoft Windows API, Pthread,Sun RPC
Configuration	J2EE and EJB configuration files, ASP.NET, and BEA WebLogic™ configuration files
Java	JMS, JNDI, J2EE, Apache Commons, Log4J, ORO, Struts, ATG Dynamo™, Hibernate, Spring, and iBatis Google Web Toolkit (GWT). Direct Web Remoting (DWR)
JSP	JSTL Core, JSTL SQL, and Apache Struts HTML EL
SQL	MOD PLSQL

一般的に良く使われるサードパーティーライブラリーの使い方もチェックすることができます

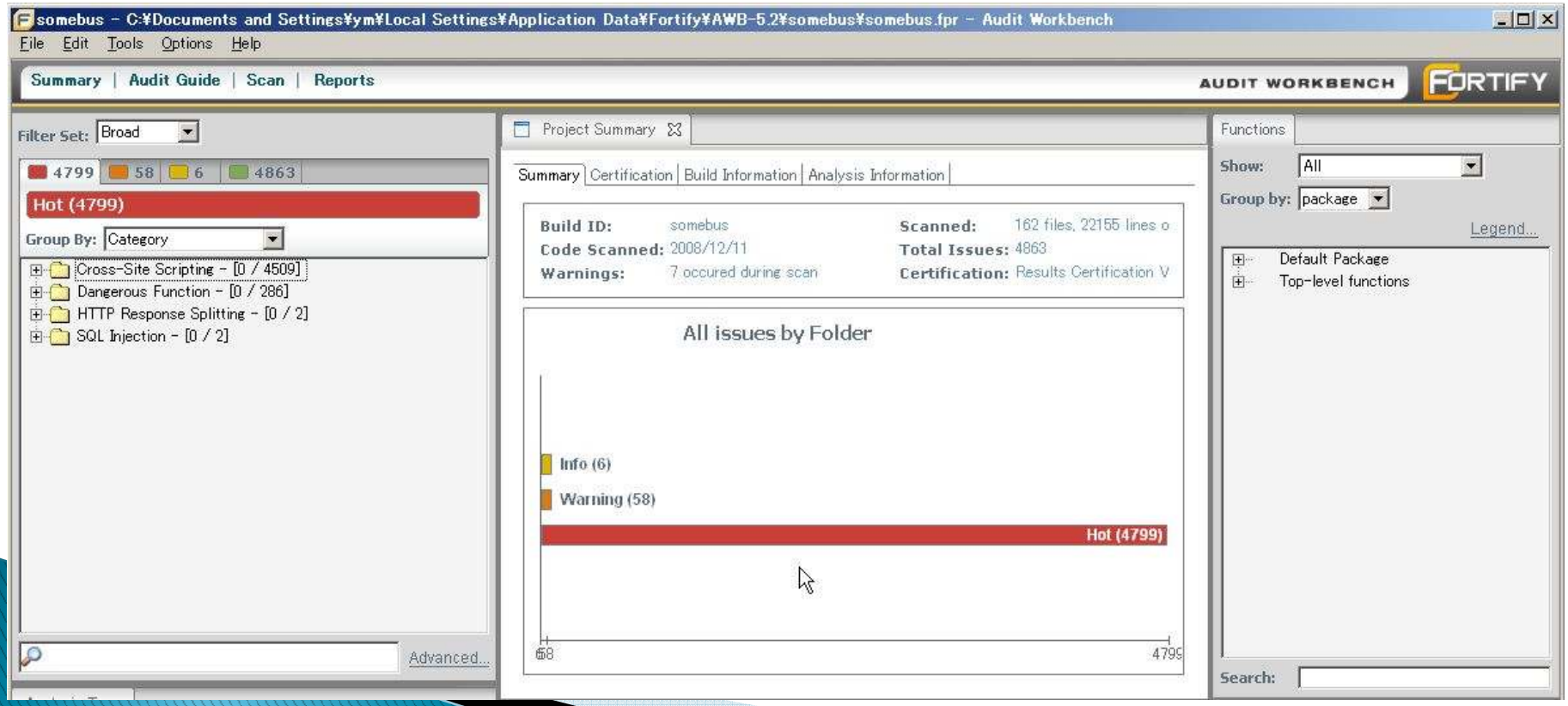
攻撃、バグ、コーディングガイドライン

Fortify SCA
Audit WorkBench



検査実験　稼働中Webシステムで検査

- ある企業の公開サイト
  - コンシューマ向け販売および決裁を実施
  - 後処理も実施



検査実験　稼働中Webシステムで検査

発見項目	発見数
Cross-Site Scripting	4509
Dangerous Function	286
HTTP Response Splitting	2
SQL Injection	2

- HTTP Response SplittingとSQL Injection
  - フレームワークの利用により、該当処理がまとめられ非常に少ない。一部、イレギュラで追加したコードで問題を検出した。
- Dangerous Function
  - 古いPHPバージョンで開発しているため、問題のある関数を使用している。
    本件は、主にMySQL関数が該当している。
- Cross-Site Scripting
  - 注意が行き届いていない。広範囲に問題を検出した。修正とテストに大量の工数が必要と思われる。

PHPの総合商社

ゼンド・ジャパン株式会社
Zend Japan, Ltd.
取締役 佐藤栄一 satou@zend.co.jp
〒150-0002 東京都渋谷区渋谷3-3-5
NBF渋谷イーストト4F
TEL 03-5766-1150 FAX 03-5766-1153
Webサイト http://www.zend.co.jp/