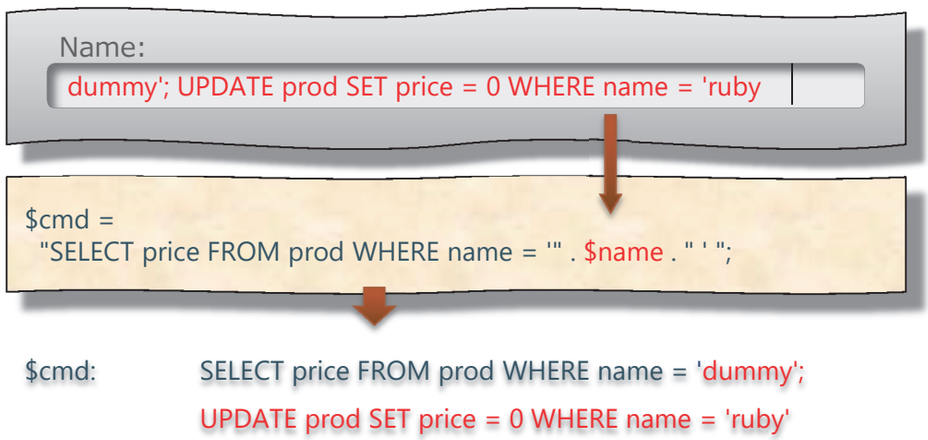
情報システムへのインジェクションアタックに対する包括的な防御手段

電子情報システム専攻　コンピュータ・アーキテクチャ研究グループ
塩谷　亮太

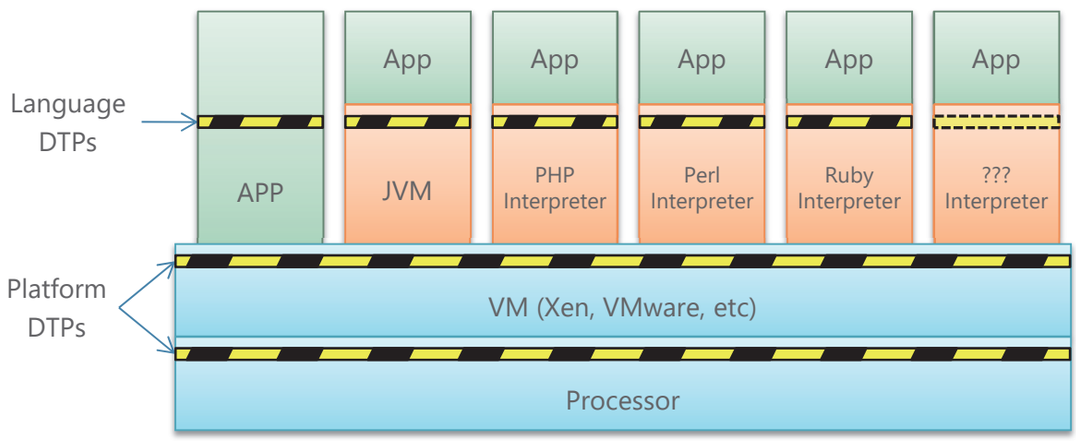
- 背景：インジェクション・アタック：
  - ◇ アプリの脆弱性を突き，
    - □ プログラムの入力に攻撃コードを与えて（インジェクション），
    - □ プログラマの意図に反する動作を起こさせる

SQL インジェクション：



- 問題：いたちごっこ になる
  - ◇ 既存技術のほとんどは，パターン・マッチングなどの事後対策型アプローチしかとれていない

- DTP (Dynamic Taint Propagation)
  1. 外部から入力されたデータにテイント（汚染）というマークをつける
  2. プログラムの実行に従いテイント情報を伝搬させる
  3. テイント情報のついたデータが危険な操作に使われたら攻撃を検出
     - □ 例）システムコール，SQLコマンド

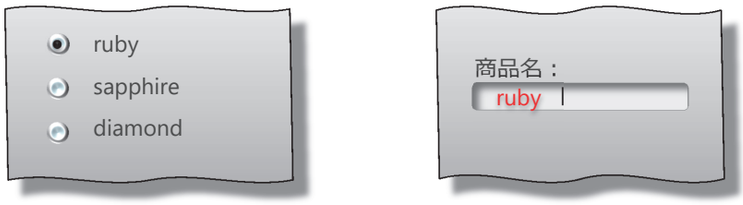


情報システムへのインジェクションアタックに対する包括的な防御手段

電子情報システム専攻　コンピュータ・アーキテクチャ研究グループ
塩谷　亮太

■ 既存のDTPの問題点
- ◇ テーブルの参照を使う時，誤検出と検出漏れのトレード・オフに陥る
  - □ テーブルの参照　：　$o = table[$i];
    1. 安全とみなすと ⇒ セキュリティー・ホールを生む
    2. 危険とみなすと ⇒ 大量の誤検出を生む
- ◇ 既存のDTPの多くは前者を選択
  - □ ローカルな視点でテイント情報を伝搬させており，マクロな構造に着目できていない

■ 本質：何を追ってテイント情報を伝搬させるべきか？

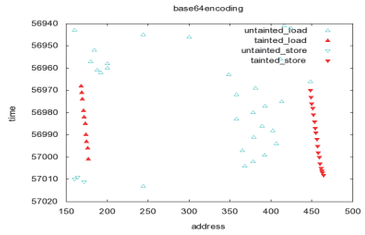


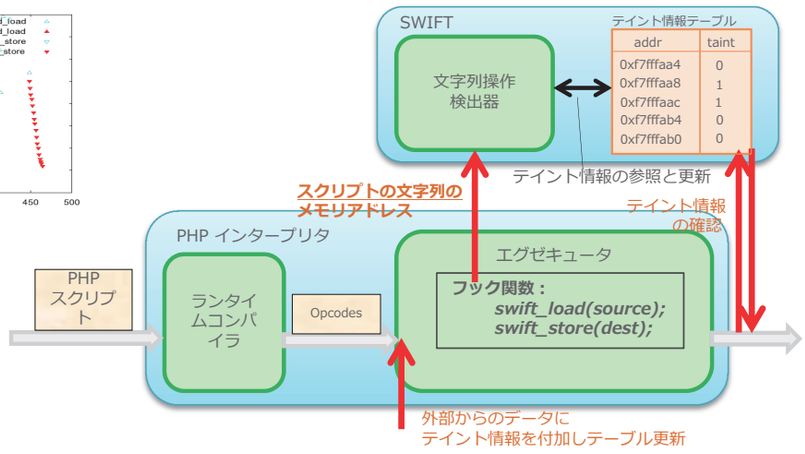
ラジオボタン：SAFE
プログラマが用意した文字列からの選択

テキストボックス ：UNSAFE
ユーザが任意の文字列を入力

- ◇ テキストボックス型の文字列操作だけを追跡しテイント情報を伝搬　+ コマンド解析による検出

■ SWIFT
- ◇ Load/Store 命令のアドレス・トレースのみに着目する
  - □ そこから，文字列操作を識別し，load 文字列から store 文字列へとテイント情報を伝搬させる
- ◇ 包括的なインジェクション・アタックへの対処が可能に
- ◇ 現在はプロセッサ（シミュレータ）と，PHP 上にそれぞれ実装





■ 検出の例：WordPress
- ◇ WordPress-3.0.1, http://www.exploit-db.com/exploits/15684/
- ◇ Spider Event Calendar v1.3.0(WordPress プラグイン), http://www.exploit-db.com/exploits/25723/

◇ INFORMATION PROCESSING DEVICE, INFORMATION PROCESSING METHOD, AND COMPUTER, 米国特許番号：US 8413240 B2 (2013).
◇ 情報処理装置、情報処理方法及びこれを実現させるためのプログラム, 特許第4669053号 (2011).