Pixy - analiza statyczna skryptów PHP

Adam Morawski

MIMUW

4 kwietnia 2012

Co to jest?

- Analiza programu bez jego uruchamiania
- Zazwyczaj pod konkretnym kątem
- Zazwyczaj dużo szybsza i prostsza do uruchomienia od testowania czy weryfikacji wszystkich możliwych przebiegów programu
- Wspomaga na bieżąco proces tworzenia oprogramowania
- Kompilatory i zewnętrzne narzędzia

Co można analizować?

- zgodność typów
- niezainicjalizowane zmienne i nieużywane przypisania
- odwołania do złych miejsc w pamięci (null pointer, poza zakres)
- nieprawidłowe korzystanie z funkcji bibliotecznych, zasobów
- możliwe wartości zmiennych (analiza przepływu danych), osiągalność bloków kodu, szukanie miejsc podatnych na atak
- styl programowania (metryki itp)
- mniej lub bardziej czuła na przepływ sterowania
- zazwyczaj może się mylić w obie strony

Zagrożenia w skryptach PHP

- Dane pochodzące od użytkownika
- SQL injection
- Cross-site scripting itp
- include/eval czegokolwiek
- Możliwość uruchomienia innego skryptu, niż zakładał autor
- register_globals

Jak sobie radzić z zagrożeniami?

- Nieprzekazywanie danych użytkownika bezpośrednio do bazy danych/wyświetlenia/include/eval
- Funkcje oczyszczające dane (htmlspecialchars, addslashes)
- Ostrożna konstrukcja zapytań do bazy
- Wyłączenie register_globals
- Wyłączenie wyświetlania błędów/ostrzeżeń w wersji produkcyjnej :)

Czemu to może być trudne?

- Brzydki kod, z funkcjonalnościami dodawanymi przez lata przez przypadkowych autorów
- Obrabianie danych wyświetlanych/przekazywanych do bazy rozproszone po kodzie
- Spaghetti code, na dodatek przemieszany z HTML, JavaScript itd
- Dużo includowanych plików, które coś robią a nie tylko definiują funkcje
- Przepisanie wszystkiego od nowa może być nieopłacalne

Pixy

- analiza statyczna skryptów PHP pod kątem zagrożeń XSS i SQL injection
- napisany w javie
- analiza przepływu danych od źródeł (użytkownik/baza danych) do ujść (baza danych/wyświetlanie danych na stronie/inne groźne funkcje)
- http://pixybox.seclab.tuwien.ac.at/

Działanie Pixy

- parsowanie w oparciu o orginalną gramatykę PHP, tłumaczenie do P-tac i grafu przepływu sterowania
- analiza ”zanieczyszczenia” danych, od źródeł do ujść
- opis zachowania się funkcji wbudowanych
- flow-sensitive, interprocedural, context sensitive
- analiza literałów: do obsługi include’ów, indeksowania tablic
- analiza aliasów ($a =& $b)
- brak obsługi obiektowości

Analiza literałów

- Krata przenoszonych wartości + funkcja przejścia
- reprezentacja dowolnych literałów, obsługa tablic w miarę możliwości

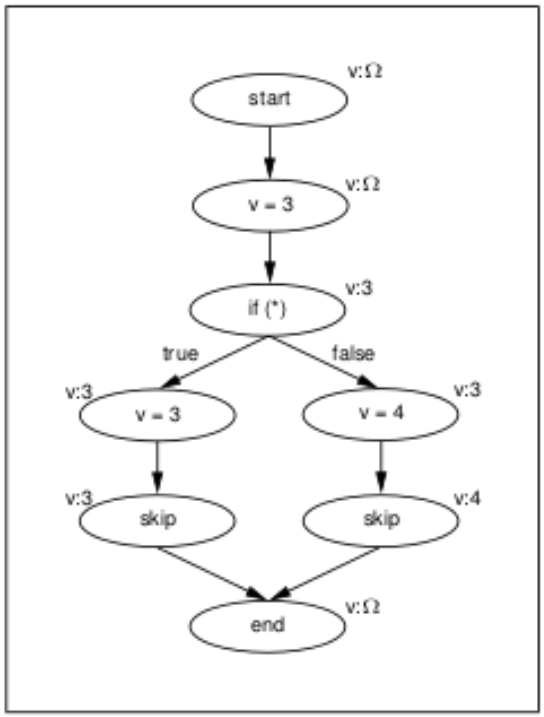

Fig. 2. Example CFG with associated analysis information.

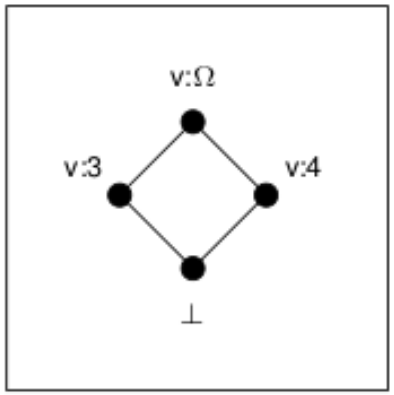

Fig. 3. A simple lattice.

Analiza literałów - tablice

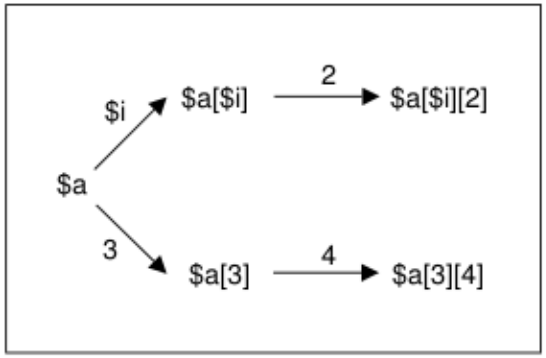

Fig. 6. Array Tree Example.

```
strongOverlap(Variable target, Place source) {
  if source known as array
    if target known as array
      for all direct elements of target
        if there is a direct element of source with the same direct index
          strongOverlap(target element, source element)
        else
          set the array tree of the direct element to Omega
  else if source is literal or constant
    set the array tree below target to NULL
  else
    set the array tree below target to Omega

  set the target literal to the source literal
}
```

Fig. 7. Strong Overlap Algorithm.

Analiza aliasów

- nie musi być włączana
- pozwala stwierdzić które zmienne mogą lub muszą wskazywać na dane przypisane do innych zmiennych
- po przeprowadzeniu analizy aliasów odkrycia nanoszone na diagram przepływu dla literałów

Left Variable	Literal Analysis	Taint Analysis
Not an array element and not known as array ("normal variable").	strong update for must-aliases, weak update for may-aliases	strong update (taint, CA flag) for must-aliases, weak update (taint, CA flag) for may-aliases
Array, but not an array element.	strong overlap	target.caFlag = source.caFlag; strong overlap (taint)
Array element (and maybe an array) without non-literal indices.	strong overlap	target.root.caFlag ⊔= source.caFlag; strong overlap (taint)
Array element (and maybe an array) with non-literal indices.	weak overlap for all MI variables	target.root.caFlag ⊔= source.caFlag; weak overlap (taint) for all MI variables

TABLE II

ACTIONS PERFORMED BY LITERAL ANALYSIS AND TAINT ANALYSIS FOR SIMPLE ASSIGNMENT NODES DEPENDING ON THE LEFT-HAND VARIABLE.

Analiza żanieczyszczeń"

- dane mogą być zanieczyszczone lub nie w odpowiednim kontekście (XSS/SQLi)
- analiza przepływu z uwzględnieniem informacji zdobytych w poprzednich analizach
- modele funkcji wbudowanych mogących wpływać na zanieczyszczenie danych:
  - strong sanitization
  - weak sanitization
  - multi-dependency
  - inverse multi-dependency
  - evil
- opis w których zmiennych środowiskowych są bezpieczne dane, a w których nie (przy analizie SQLi - 2 poziomy czystości) + flagowanie całych tablic jako czyste

Demo

Wyniki

Program	File	LOC	Variables	Vulnerabilities	FP's	Advisories
PhpNuke 6.9	Reviews Module	8409	3113	15	5	BugTraq: 10493, 10524, 365368
	YourAccount Module	9070	3452	9	25	BugTraq: 13007, 394971, 394867, 321324
PhpMyAdmin 2.6.0-pl2	select_server.lib.php	89	23	9	0	PMASA-2005-01
Gallery 1.3.3	search.php	1810	530	2	1	BugTraq: 348514
	login.php	1719	488	1	0	BugTraq: 8039
	Totals	21097	7606	36	31	

Table 1. Known vulnerabilities discovered by Pixy.

Program	File	LOC	Variables	Vulnerabilities	FP's	Advisories
Simple PHP Blog 0.4.5	preview_cgi.php	6938	2342	3	5	TUVSA-0511-001, BugTraq 415463
	preview_static_cgi.php	6883	2316	4	4	
	colors.php	6971	2313	1	6	
Serendipity 0.8.4	personal.inc.php	6588	2305	2	1	TUVSA-0509-001, BugTraq 412023
Yapig 0.95b	view.php	5128	1302	5	0	TUVSA-0510-001, BugTraq 413255
	Totals	29508	10578	15	16	

Table 2. Unknown vulnerabilities discovered by Pixy.

Zalety i wady

- Narzędzie sprawdziło się w praktyce
- Wymaga stosunkowo niewielkiego nakładu pracy
- Łatwe dostosowanie modeli do własnych funkcji oczyszczających dane itp
- Dość precyzyjne
- Nie obsługuje obiektowości (ze względu na wydajność)
- Nie uznaje oczyszczeń przez wyrażenia regularne
- Nie wnioskuje o powiązanych ze sobą rozgałęzieniach kodu
- Możliwe pomyłki w obie strony
- Nie radzi sobie z polskimi znaczkami, początkiem xmla itp

Bibliografia

1. Nenad Jovanovic, Christopher Kruegel, Engin Kirda: *Pixy: A Static Analysis Tool for Detecting Web Application Vulnerabilities (Technical Report)*
2. Nenad Jovanovic, Christopher Kruegel, and Engin Kirda: *Precise Alias Analysis for Static Detection of Web Application Vulnerabilities*
3. http://pixybox.seclab.tuwien.ac.at/pixy/