Zapewnienie bezpieczeństwa w całym cyklu życia aplikacji

(czyli dlaczego lepiej zapobiegać chorobom, niż leczyć je w zaawansowanym stadium)

dr inż. Jakub Botwicz
CISSP, ECSA

jakub.botwicz@pl.ey.com

OWASP

9.10.2012

The OWASP Foundation

http://www.owasp.org

Zapewnienie bezpieczeństwa w całym cyklu życia aplikacji

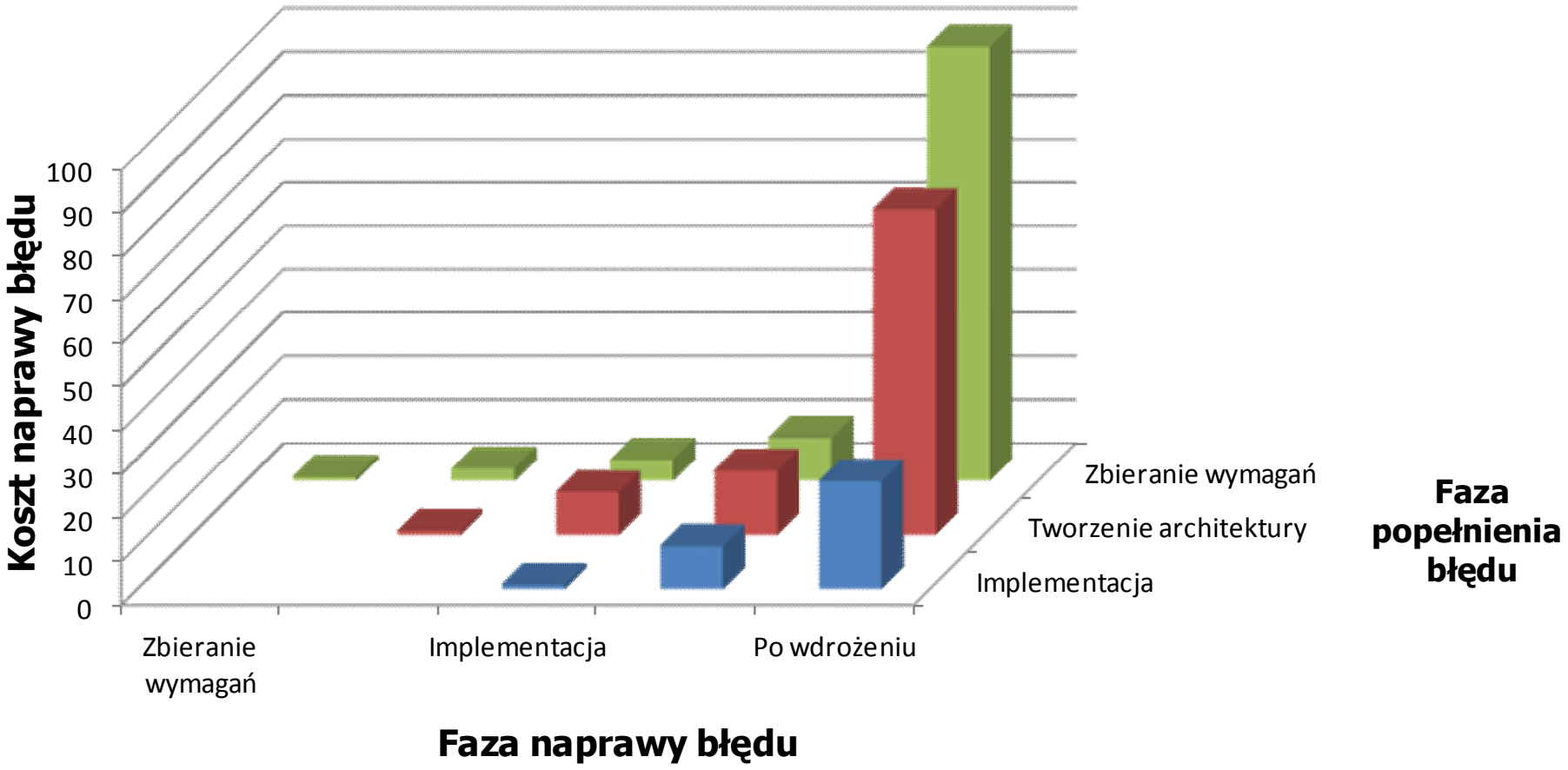
czyli dlaczego lepiej zapobiegać chorobom, niż leczyć je w zaawansowanym stadium

- Koszty naprawy błędów na różnych etapach życia aplikacji
- Projekty:
  - OWASP SAMM (Software Assurance Maturity Model)
  - BSIMM (Building Security In Maturity Model)
- Wybrane aktywności:
  - Szkolenia
  - Specyfikacja wymagań bezpieczeństwa
  - Testy automatyczne
  - Użycie komponentów zewnętrznych

Motywacje

- Przypuśćmy, że w naszej aplikacji został znaleziony błąd po wdrożeniu produkcyjnym
- Jakie są konsekwencje?
  - Straty reputacyjne
  - Utracone zyski (przerwy w działaniu aplikacji lub błędne działanie aplikacji)
  - Koszty poprawek projektu, kodu, ponownego testowania i wdrożenia
- Czy można było tego uniknąć?

Koszty naprawy błędów w aplikacjach



Źródło: McConnell, Steve (2004). Code Complete. Microsoft Press.

Motywacje

- Im wcześniej wykryjemy i usuniemy błędy tym mniejsze poniesiemy koszty z nimi związane
- Profilaktyka choć wymagająca dodatkowych nakładów, w efekcie okazuje się być opłacalna
- W jaki sposób można tworzyć aplikacje z mniejszą liczbą błędów oraz szybciej wykrywać błędy?

BUILDING SECURITY IN MATURITY MODEL

Building Security In Maturity Model v.4

- Badanie praktyk w dziedzinie bezpieczeństwa przeprowadzone w 51 firmach:
  Google, Intel, Microsoft, VMware, Adobe, SAP, Bank of America, EMC, Nokia, Symantec, Visa
- Firmy różnej wielkości: w przedziale od 11 do 30000 programistów, średnio 4455
- Pierwsze wyniki opublikowano w 2009 roku i od tego czasu były 3-krotnie aktualizowane
- Powstał zbiór 12 praktyk ze 111 aktywnościami

BSIMM – praktyki

- Szkolenie
- Strategia i metryki
- Analiza architektury
- Polityka bezpieczeństwa i zgodność z regulacjami
- Standardy i wymagania
- Modelowanie ataków
- Projektowanie i funkcjonalności bezpieczeństwa
- Przeglądy kodu źródłowego
- Testy bezpieczeństwa
- Testy penetracyjne
- Zabezpieczenie środowiska uruchomieniowego
- Zarządzanie konfiguracją i podatnościami

BSIMM – poziomy aktywności

- Poziom 0: brak aktywności w tym obszarze
- Poziom 1: podstawowy, najłatwiejszy do wdrożenia, niezbędne niezależnie od wielkości firmy i branży

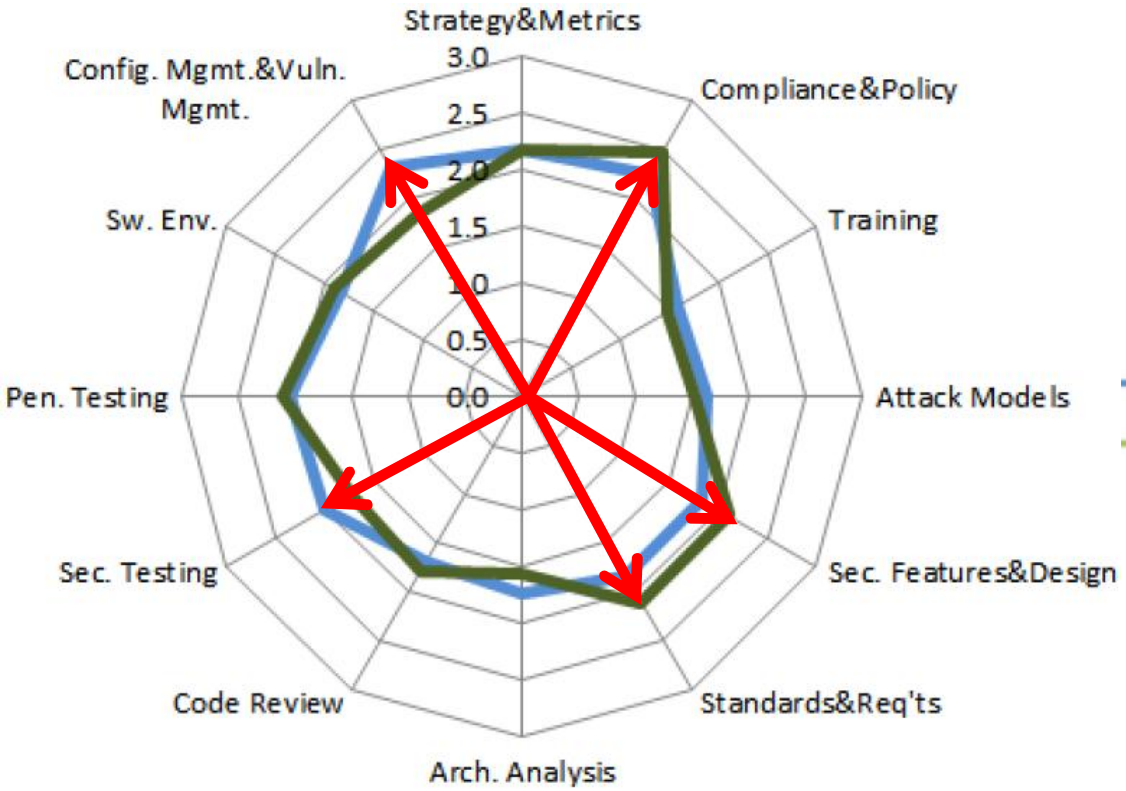
  Testowanie aplikacji przez niezależnych pentesterów
- Poziom 2: bardziej zaawansowane, dopasowane do specyfiki firmy

  Testowanie penetracyjne aplikacji okresowo z analizą pokrycia wszystkich funkcji
- Poziom 3: najbardziej kosztowne i wyrafinowane metody, na które mogą sobie pozwolić największe firmy

  Przygotowywanie własnych narzędzi do testów penetracyjnych dopasowanych do specyfiki aplikacji

BSIMM – wykorzystanie

Pozwala na ocenę i wizualizację aktualnego stanu praktyk bezpieczeństwa



(wykres radarowy: odległość od środka = poziom zaawansowania w określonej praktyce)

Źródło: Building Security In Maturity Model 4, Gary McGraw, Sammy Migues & Jacob West (2012)

BSIMM – Software Security Group

- Grupa pracowników firmy

  ekspertów w dziedzinie bezpieczeństwa
  (w różnych obszarach i stanowiskach)

- Liderzy zmian i projektów

- Przekazują wiedzę pozostałym pracownikom

- Liczba osób w SSG w badanych firmach:
  - od 1 do 100 osób
  - średnio 19,48

SOFTWARE ASSURANCE MATURITY MODEL

Software Assurance Maturity Model

- Podtytuł: *A guide to building security into software development*
- Bazuje na doświadczeniu i wiedzy autorów, a nie na wynikach przeglądu firm
- Opisuje zbiór 12 praktyk z 72 aktywnościami

SAMM – elementy opisu aktywności

- Szczegółowy opis aktywności:

A. Utilize automated security testing tools

In order to test for security issues, a potentially large number of input cases must be checked against each software interface, which can make effective security testing using manual test case implementation and execution unwieldy. Thus, automated security test tools should be used to automatically test software, resulting in more efficient security testing and higher quality results.

Both commercial and open-source products are available and should be reviewed for appropriateness for the organization. Selecting a a suitable tool is based on several factors including robustness and accuracy of built-in security test cases, efficacy at testing architecture types important to organization, customization to change or add test cases, quality and usability of findings to the development organization, etc..

Utilize input from security-savvy technical staff as well as development and quality assurance staff in the selection process, and review overall results with stakeholders.

- Rezultaty:

RESULTS

- Independent verification of expected security mechanisms surrounding critical business functions
- High-level due diligence toward security testing
- Ad hoc growth of a security test suite for each software project

Źródło: Software Assurance Maturity Model, OWASP OpenSAMM Project, Pravir Chandra (2009)

SAMM – elementy opisu aktywności

- Koszty wdrożenia:

PERSONNEL

- QA Testers (1-2 days/yr)
- Security Auditor (1-2 days/yr)
- Developers (1 day/yr)
- Architects (1 day/yr)
- Business Owners (1 day/yr)

- Wskaźniki powodzenia:

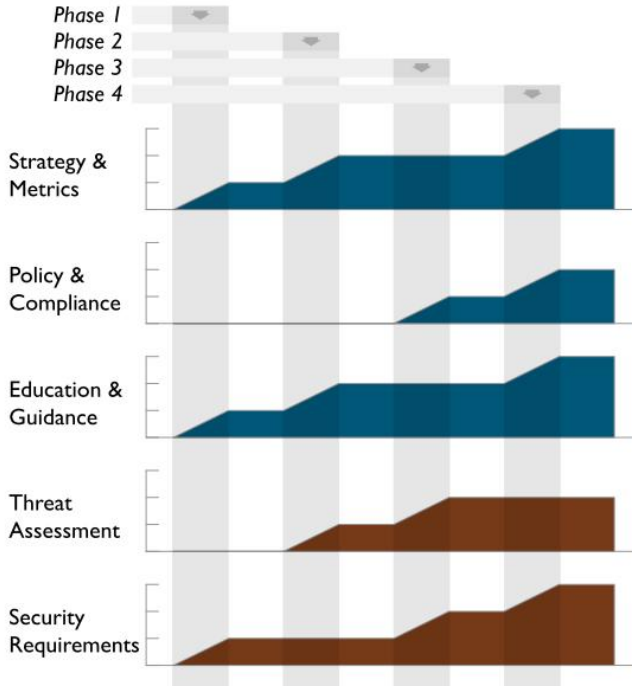
SUCCESS METRICS

- >50% of projects specifying security test cases in past 12 months
- >50% of stakeholders briefed on project status against security tests in past 6 months

Źródło: Software Assurance Maturity Model, OWASP OpenSAMM Project, Pravir Chandra (2009)

SAMM – Building Assurance Programs

- Przykładowe plany wdrożenia praktyk:
  - w jakiej kolejności wdrażać
  - które można pominąć
- Profile:
  - Dostawcy oprogramowania
  - Dostawcy usług
  - Instytucje finansowe
  - Instytucje rządowe
- Dodatkowe wskazówki:
  - Oprogramowania tworzone zewnętrznie
  - Transakcje przetwarzanie w sieci



Źródło: Software Assurance Maturity Model, OWASP OpenSAMM Project, Pravir Chandra (2009)

SAMM vs. BSIMM – porównanie

	BSIMM	SAMM
Źródło danych	Wyniki ankiety przeprowadzonej w 51 firmach	Wiedza autorów
Liczba aktualizacji	3	0
Liczba opisanych aktywności	111	72
Liczba stron	60	90
Poziom opisu aktywności	Pobieżny	Szczegółowy
Sposób wdrażania aktywności	Brak opisu	Opisany

Jak korzystać z SAMM lub BSIMM

1. Uzyskanie wsparcia od kadry zarządzającej
   - sponsorzy prac i projektów
   - decydenci w sprawach strategicznych
2. Zbadanie stanu początkowego
   jednego, kilku wybranych lub wszystkich obszarów
3. Zebranie grupy Software Security Group
4. Określenie celów i etapów ich osiągnięcia

Jak wdrażać SAMM lub BSIMM

5. Wdrożenie pilotażowe
   (w mniejszym obszarze lub części funkcjonalności)
   a) planowanie
   b) wdrożenie
   c) sprawdzenie kosztów i korzyści
   d) wyciągnięcie wniosków i doświadczeń

6. Wdrożenie pełne
   (dla całej firmy lub wszystkich aplikacji)

NAJLEPSZE PRAKTYKI

Szkolenia – najlepsze praktyki

- Dla nowych pracowników
- Powtarzane i uaktualniane okresowo (raz na rok)
- Wykorzystujące historyczne dane firmy (np. przypadki popełnionych wcześniej błędów)
- Dopasowane do stanowiska (programista, tester, architekt)
- Obejmujące również kadrę zarządzającą
- Z udostępnionymi materiałami do dalszej nauki

Wymagania dotyczące bezpieczeństwa

- Wymagania bezpieczeństwa powinny być dokładnie opisane razem z wymaganiami funkcjonalnymi
- Wymagania powinny być opisane pozytywnie
  - Aplikacja ma być odporna na ataki SQL Injection
  - Aplikacja ma walidować dane wejściowe pod względem występowania w nich znaków ze składni języka SQL i prawidłowo je kodować lub usuwać
- Defensywne aplikacje są łatwiejsze do obrony
  - Pole login może zawierać dowolne znaki testowe
  - Pole login może zawierać litery, cyfry i podkreślenia

Testy automatyczne

- Zalety testów automatycznych
  - krótszy czas wykonania niż w testów manualnych
  - możliwość sprawdzenia dużej liczby przypadków testowych
  - powtarzalność testów i wyników
- Wady testów automatycznych
  - konieczność przygotowania testów (konfiguracji narzędzi)
  - jakość wyników uzależniona od jakości testów

Testy bezpieczeństwa – najlepsze praktyki

- Aby połączyć zalety testów automatycznych i manualnych
  - Testy automatyczne:
    - wykonywać jak najczęściej np. przy każdym wprowadzeniu poprawek do systemu kontroli wersji
    - używać jako testy regresyjne
  - Testy manualne:
    - wykonywać przy istotnych zmianach w aplikacji
    - na podstawie ich wyników uaktualniać testy automatyczne

Użycie komponentów zewnętrznych

- Użycie sprawdzonego kodu zewnętrznego daje duże korzyści
  - dostajemy gotowy do użycia kod – oszczędzamy czas
  - korzystamy z pracy innych programistów i testerów – oszczędzamy koszty pracy naszych pracowników
- Ale w zamian za to
  - błędy w użytym kodzie stają się błędami naszych aplikacji
  - powinniśmy śledzić informacje o wykrytych błędach w kodzie, który używamy

PODSUMOWANIE

Podsumowanie

- Im wcześniej wykryjemy i usuniemy błędy tym mniejsze poniesiemy koszty z nimi związane
- Najlepiej uczyć się na cudzych błędach i nie popełniać własnych
- Dokumenty SAMM i BSIMM mogą służyć do sprawdzenia aktualnego stanu praktyk naszej firmy i przygotowania planu poprawy tego stanu
- Mocne strony:
  - SAMM: szczegółowy opis praktyk, podane sposoby wdrożenia
  - BSIMM: wyniki oparte na rzeczywistym badaniu praktyk, zawiera więcej aktywności

Dziękuję za uwagę!

Czy mają Państwo jakieś pytania?