UNIVERSITÀ DEGLI STUDI DI MILANO

Facoltà di Scienze Matematiche, Fisiche e Naturali

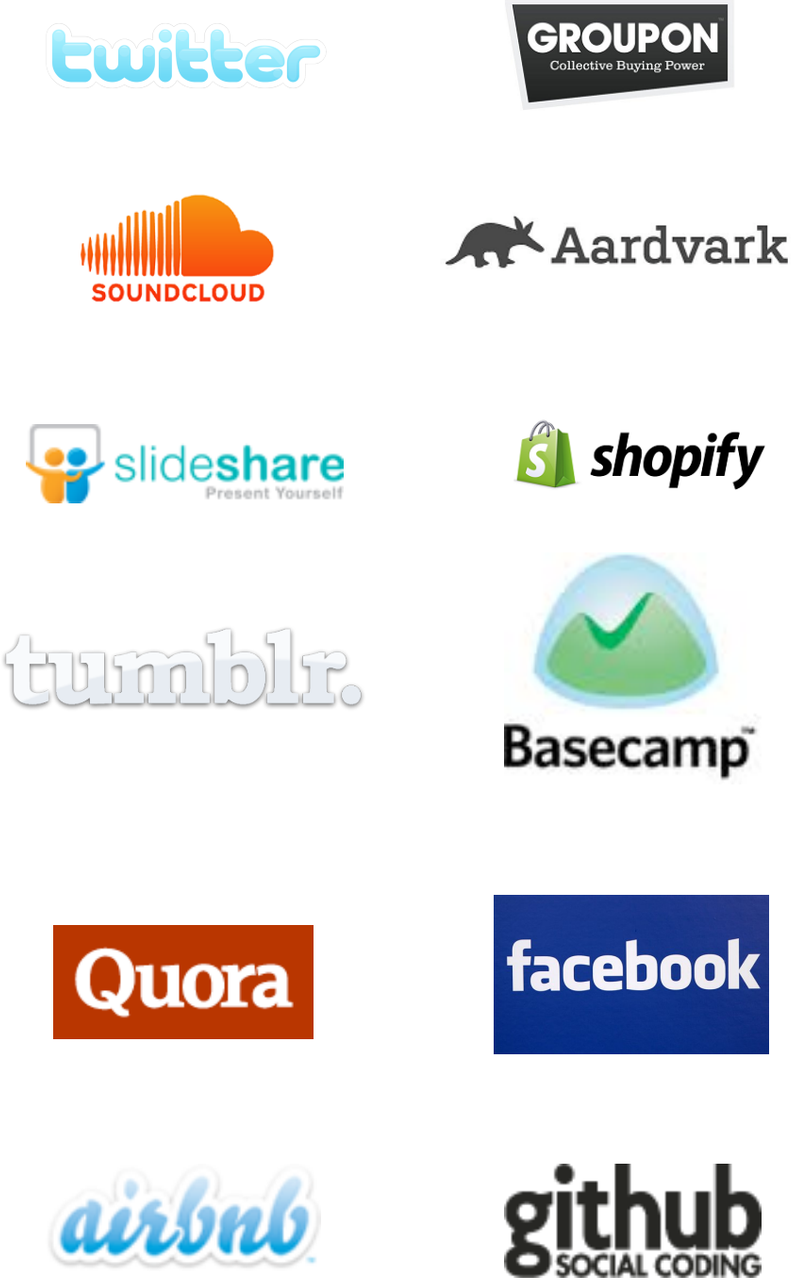
Corso di Laurea in Scienze e Tecnologie dell'Informazione

TECNICHE DI ANALISI STATICA PER LA SICUREZZA DI APPLICAZIONI WEB: PROBLEMATICHE ED IMPLEMENTAZIONI

Dario Battista Ghilardi
753708

Relatore:
Prof. Marco Cremonini

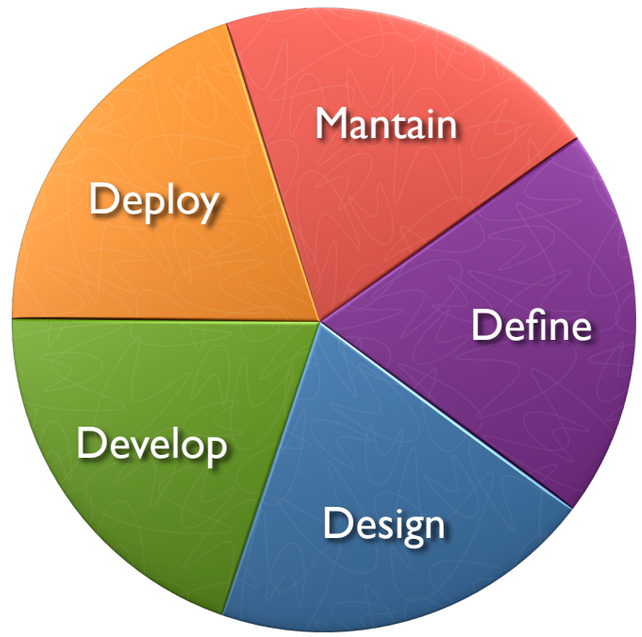
LA DIFFUSIONE DELLE APPLICAZIONI WEB E' IN CONTINUO AUMENTO



Le applicazioni necessitano di sicurezza

- Analisi statica
- Analisi dinamica
- Code Review
- Penetration testing

ESECUZIONE



ma la sicurezza è un processo...

anticipare i controlli sulla sicurezza -> usare l'analisi statica durante lo sviluppo

PROBLEMA:

- Costo di correzione elevato

-

In questa tesi:

- Stato dell'arte sull'analisi statica
- Strumenti e tools
- Proposta di un nuovo tool

Vantaggi:

- la correzione costa di meno
- eliminazione della finestra di esposizione
- lo sviluppatore è coinvolto consapevole

Analisi statica:

- controllare codice senza eseguirlo

In PHP, come si fa?

Diversi livelli:

analisi codice sorgente

analisi tokens

analisi ast

analisi control flow graph

analisi bytecode

Stato dell'arte su PHP:

- Codesecure
- Fortify 360

(proprietari, implementazione non disponibile)

- Pixy
- RIPS

(open source)

Pixy:

- eurecom, in java
- Analisi data flow
- Usa il control flow graph
- abbandonato dal 2007
- notevoli falsi positivi
- Non supporta PHP 5, quindi non usabile per ora

Problema: come valuto validazioni custom?

Saner!

RIPS:

- Johannes Dahse
- Usa tokenizer
- definisce sinks, sanitization functions, parametri, valori tainted a priori
- in sviluppo
- Non supporta completamente PHP 5
- Codebase mal strutturata
- Veloce, molti falsi, buona gui

Soluzioni esistenti non soddisfacenti:

progettato Vulture in collaborazione con EURECOM

- basato su symfony framework
- scritto in php
- estensibile
- parsing ast generato da PHP-Parser
- definizione di sinks, sanitization functions
- previsto supporto PHP 5
- attualmente in sviluppo, non ancora usabile

Sommario:

- introduzione analisi statica nel ciclo di sviluppo software

-> lo sviluppatore controlla man mano che scrive codice

Considerazioni:

- tool open non hanno catturano audience
- sviluppo del tool su linguaggi dinamici come php complesso
- tool commerciali non sono stati realizzati per essere usati durante lo sviluppo ma come supporto alla code review

quindi

- ramo quasi inesplorato e promettente
- esigenza sentita -> business di sicuro successo

Sviluppi futuri:

- Portare a termine funzionalità Vulture
- estendere Vulture con plugin

Domande?

Grazie.