# PreCVE: Rilevamento proattivo di vulnerabilità applicative tramite SAST ed Intelligenza Artificiale


*Emanuele De Lucia*

# Table of contents_

# 01

# Introduzione: Progetto PreCVE.dev

# Sfide e Complessità

Sapevate che :

- gli strumenti SAST (Static Application Security Testing) tradizionali generano un grande numero di avvisi ?
- i tool tradizionali faticano a comprendere l'impatto reale di una debolezza nel contesto dell'intera applicazione ?
- i Large Language Models (LLM), seppur «abili» nell’analisi del codice, sono piuttosto inclini ad allucinazioni e a falsi positivi/negativi se usati inadeguatamente (es. "zero-shot") ?

Sfida:

Come possiamo unire la precisione dell'analisi statica con la comprensione semantica degli LLM per creare un sistema che trovi solo ciò che conta ?

# Sistema Ibrido di "Code Intelligence"_

## Fase 1: Indicizzazione Profonda (Offline)

Trasformiamo il codice sorgente in un *grafo semantico* arricchito, non solo in *embedding* vettoriali. Questo ci permette la creazione di una mappa interconnessa dove non solo comprendiamo *cosa fa* ogni funzione, ma anche *come interagisce* con le altre e, soprattutto, *quale rischio per la sicurezza* questa interazione comporta.

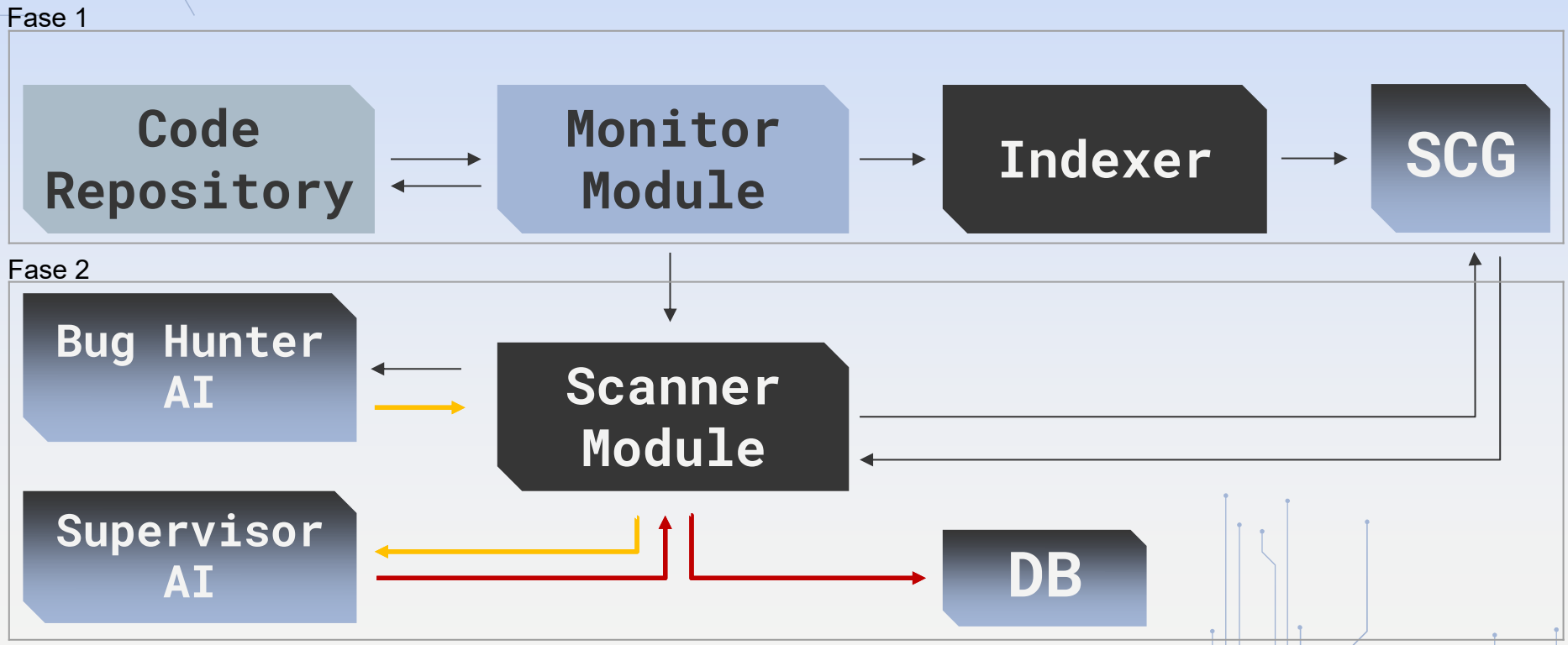
## Fase 2: Giudizio AI (Online)

Approccio che sfrutta il grafo per analizzare *nuovi commit* e/o progetti con contesto di codice specifico seguito da un processo di «*sfida*» fra due AI Fine-Tuned (per formato di *output*, *risparmio token*, *inferenza* più rapida e conoscenza specifica *framework* es. *$wpdb come oggetto globale WP per interazione con DB*) + Retrieval-Augmented Generation (RAG).

# 02

# Architettura

# Architettura



# 03

# Indicizzatore Semantico

# Indicizzatore Semantico_

## Analisi Taint Inter-Procedurale

L’analisi non si limita alla singola funzione (Intra-Procedurale) ma di tipo Inter-Procedurale
(obiettivo: **contesto per AI**)

## Tracciamento «Sources» e «Sinks»

Verifica di come una “Sources” possa propagarsi fino a raggiungere funzioni pericolose «Sinks»

## Concetto di “WorkList”

I blocchi «contaminati» vengono aggiunti ad una Worklist. Il sistema itera: estrae un blocco, vede quali funzioni chiama (callees) e propaga la contaminazione fino a raggiungere il «punto fisso».

# Indicizzatore Semantico_

## Obiettivo:

### Trasformare questo

```php
// File: src/api/routes.php

class APIRoutes {
    public function handle_user_update($user_id) {
        $data = $_POST['userdata']; // Fonte di Taint
        $clean_data = sanitize_user_data($data);

        $user = new User();
        $user->update($user_id, $clean_data); // Sink potenziale
    }
}
```

### in questo


```json
{
  "file": "src/api/routes.php",
  "name": "handle_user_update",
  "type": "method_declaration",
  "start_line": 5,
  "end_line": 11,
  "caller_ids": "[\"src/core/router.php::dispatch_route::88\"]",
  "callee_ids": "[\"src/utils/sanitizer.php::sanitize_user_data::23\",
  "parents_by_name": "[\"APIRoutes\"]",
  "parent_ids": "[\"src/api/routes.php::APIRoutes::3\"]",
  "taint_summary": "{\"sources\": [\"_POST\"], \"sinks\": [\"update\"],
  "token_count": 52
}
```

# Indicizzatore Semantico_


*Attributo del nodo*: "file", "name", "type", "start_line", "end_line".

*Caller ids*: arco entrante del nostro nodo. Dispatch_route alla riga 88 chiama handle_user_update.

*Callee ids*: arco uscente del nostro nodo. handle_user_update chiama sanitize_user_data e update.

*Parent ids*: arco gerarchico. handle_user_update appartiene alla classe APIRoutes.

*Taint Summary*: Contesto Arricchito. Informa l'AI che il nodo introduce dati non fidati.

# 04

# AI vs AI

# AI vs AI_

## Obiettivo:

Migliorare drasticamente la correttezza delle segnalazioni “sfidando” le conclusioni della prima AI mediante una seconda AI.

### «Bug Hunter» AI Prompt

```
SYSTEM_PROMPT: You are a principal
code security analyst...

USER_PROMPT: ...analyze the
security implications of the
PATCH...

FILE UNDER REVIEW: src/user/api.php
........
```

### «Supervisor» AI Prompt

```
SYSTEM_PROMPT: You are a Senior
Security Analyst acting as a
skeptical peer reviewer.

USER_PROMPT: …your mission is to
**verify or refute** this finding
with extreme prejudice…

**STRICT DIRECTIVES:** ... your
primary goal is to find flaws in
the initial analysis…
```

# AI vs AI_

## Processo di Revisone Contraddittorio:

La combinazione sinergica di due prompt crea un processo di revisione contraddittorio.
Al supervisore viene detto di essere **estremamente scettico** (*with extreme prejudice*) e gli vengono forniti **solo i fatti concreti** (*code & patch*).
Questo lo costringe a basare la sua decisione finale unicamente sulle prove tangibili e rappresenta un'efficace strategia per eliminare eventuali falsi positivi generati dalla prima analisi.

# 05

# Findings

# Findings_1

## curl

: curl/curl

**COMMIT_SHA**: e3c06.....

**FILE**: lib/vquic/vquic.c

**TYPE**: CWE-703: Incorrect Check for Unusual Condition

**CONFIDENCE**: Medium (Fix)

**AI EXPLANATION**: The original code used an assertion to ensure all bytes were sent, which could lead to undefined behavior if the condition was not met. The patch improves robustness by checking the condition and handling it explicitly with an error message and return code.

**TIMESTAMP**: 2025-09-09 06:20:27

**COMMIT_TIMESTAMP**: 2025-09-08 14:05:27

# Findings_2

## FFmpeg

:FFmpeg/FFmpeg

**COMMIT_SHA**: 7321e.....

**FILE**: libavformat/img2dec.c

**TYPE**: CWE-120: Buffer Copy without Checking Destination Buffer Size

**CONFIDENCE**: High (Vulnerability)

**AI EXPLANATION**: The original code used a fixed-size buffer (`filename_bytes`) to store the filename. If the filename from the input (`s1->url`) exceeded the size of this buffer, a buffer overflow could occur. The patch addresses this by using `AVBPrint`, which dynamically allocates memory to accommodate the filename, preventing the overflow. The patch also removes the direct assignment of `s->path` which could lead to issues if `s1->url` is larger than `s->path`.

**TIMESTAMP**: 2025-09-06 07:36:14

**COMMIT_TIMESTAMP**: 2025-09-05 19:05:13

# Findings_3

## php-src

:php/php-src

**COMMIT_SHA**: 1869e.....

**FILE**: main/streams/streams.c

**TYPE**: CWE-404: Improper Resource Shutdown or Release

**CONFIDENCE**: High (Fix)

**AI EXPLANATION**: The patch addresses a resource management issue where buckets might not be properly released when filters return PSFS_FEED_ME or PSFS_ERR_FATAL. This could lead to memory leaks or resource exhaustion over time.

**TIMESTAMP**: 2025-09-10 04:04:49

**COMMIT_TIMESTAMP**: 2025-09-08 17:10:46

# Findings_4

## awesomemotive/all-in-one-seo-pack

:awesomemotive/all-in-one-seo-pack (WordPress Plugin)

**COMMIT_SHA**: 95c252e….

**FILE**: app/Common/Api/PostsTerms.php

**TYPE**: CWE-285: Improper Authentication / CWE-862 (Missing Authorization)

**CONFIDENCE**: High (Security Fix)

**AI EXPLANATION**: The patch fixes a potential vulnerability by adding a capability check to ensure that the user is allowed to access the data for the specified post.

**TIMESTAMP**: 2025-09-05 14:59:11

**COMMIT_TIMESTAMP**: 2025-08-29 16:57:29

# Findings_5

## openssl/openssl

:openssl/openssl

**COMMIT_SHA**: cb7da.....

**FILE**: ssl/record/rec_layer_d1.c

**TYPE**: CWE-190: Integer Overflow

**CONFIDENCE**: High (Security Fix)

**EXPLANATION**: The original code cast 'len' (a size_t) to an int before assigning it to '*written'. This could lead to an integer overflow if 'len' exceeded the maximum value representable by an int, potentially causing incorrect values to be written. The patch removes the cast, directly assigning 'len' to '*written', which is of type size_t, thus preventing overflow. **<- CORRETTO MA FUORI CONTESTO. DI FATTO INAPPLICABILE. (Falso Positivo, 2^14 byte == 16KB -> dimensione massima del record DTLS in plaintext)**

**TIMESTAMP**: 2025-09-11 11:47:37

**COMMIT_TIMESTAMP**: 2025-09-09 07:51:32"

# 06

# Applicazioni Pratiche

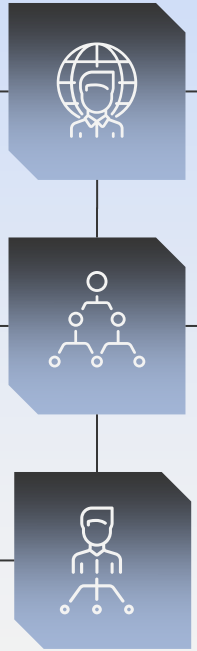
# Applicazioni_Pratiche_

## DevSecOps

Sicurezza Continua nella CI/CD

## Sec Team

Threat Hunting su Larga Scala



## M&A

Valutare la sicurezza di un codebase sconosciuto

## Offensive

Comprendere codebase complesso.

Trovare nuove vulnerabilità (0-day).

Scrivere exploit su alert di bug fixing.

## Supply Chain

Analisi dipendenze critiche

# Thanks ! _

Domande ?

**Emanuele De Lucia**

Chief Intelligence Officer @Meridian Group

*email@emanueledelucia.net*

*https://www.linkedin.com/in/emanuele-de-lucia/*

*https://www.precve.dev/*