IMPLEMENTATION OF AN INTERNAL SECURE CODING APPROACH

Company Overview

- Mid-Size Company
- Development and Design of Digital Applications
- Software Products and Solutions Deployment to several hundred customers

The Context

The Chief Information Security Officer (CISO) has an external Pentest carried out once a year to assess the security of the applications which are developed by the development teams.

Pentest results are not always used, and he wishes to enrich his approach before going into production. Therefore he has decided to set up a continuous process of fixing vulnerabilities that appear as developments progress.

The Challenge

The CISO wishes to implement an internal secure coding approach. He has decided to provide the development teams with a tool that will help to manage the applications security debt.

The Solution

To go further than a "simple" detection of problems, we proposed to equip the developers with our **Yag-suite, an all-in-one tool which enables 3 actions**:

1. Launch regular automatic code scans to detect vulnerabilities
2. Obtain a diagnostic of potentially critical vulnerabilities
3. Get remediating recommendations delivered from the source code properties

This is possible thanks to the advanced **Code Mining** technology which is natively integrated into the Yag-suite.

The tool has been especially designed to provide key functionalities which can be accessed via a simple and intuitive interface.

As it is integrated into the software forge and the IDE, the tool enables the fast implementation of secure coding best practices and facilitates the upgrading of the developers' expertise.

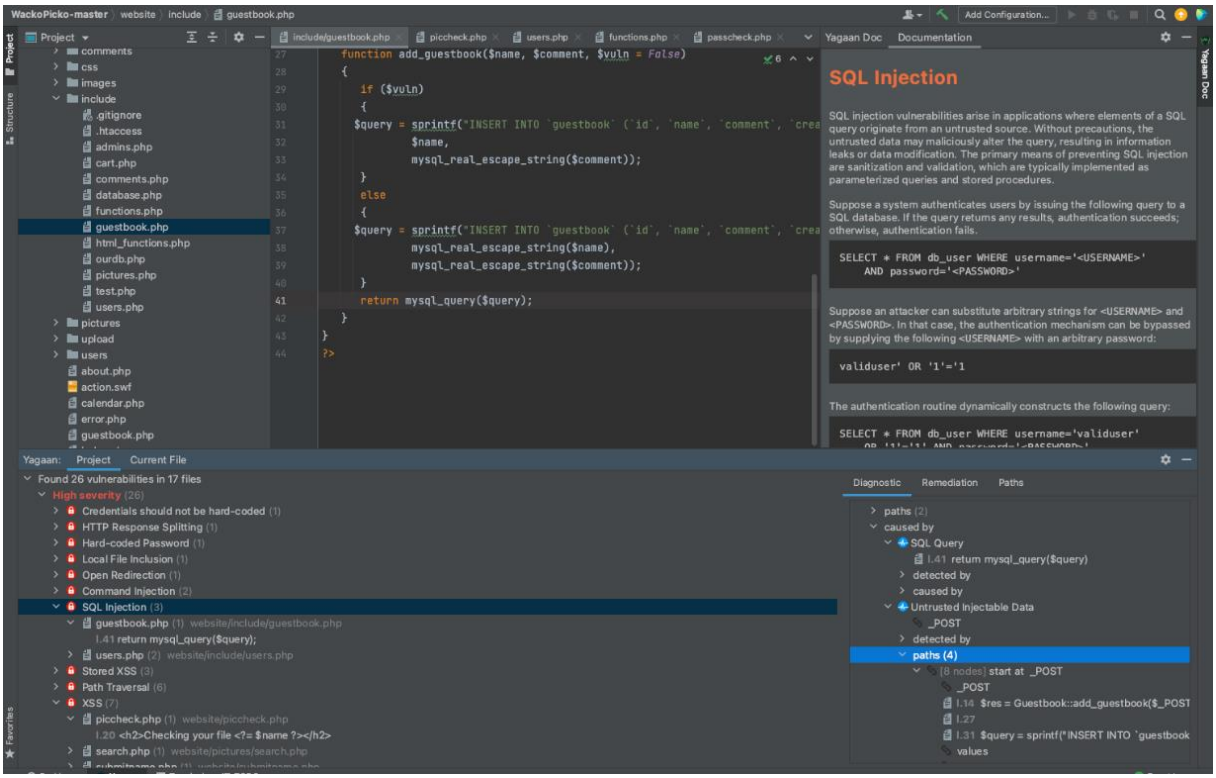
Code Mining provides an exhaustive approach

- Understand Warnings

  The tool offers the developer **a feature that renders the entire detection logic** in a detailed and educational explanation of each vulnerability.

  This explanation of the results consists of **a generic documentation of the problem** as well as **a dynamic diagnostic restoring all the information extracted from the code**, and which led to the detection of the vulnerability.

  Example of SQL injection diagnosis presented by the Yag-suite



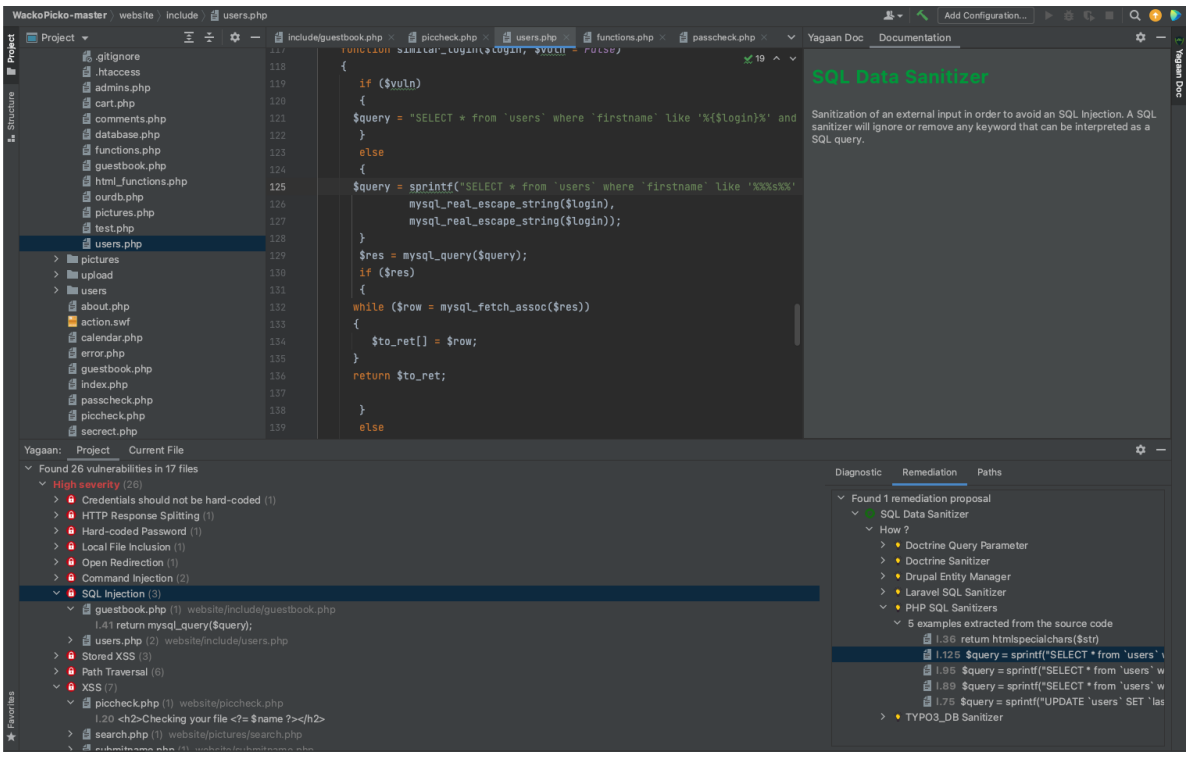
- Fix Vulnerabilities

The tool offers the developer **a contextual remediation support**.

Not only this feature presents the traditional generic documentation of possible countermeasures, but it goes much further by offering **dynamic remediation recommendations** that show code sequences extracted from the application itself.

These recommendations make it possible **to correct the vulnerability** and at the same time **avoid redeveloping countermeasures** already present in the application.

Example of a contextual remediation proposed by the Yag-suite to correct SQL injection:



Outcomes

For the CISO

For an perfectly well-optimized investment in time and money, he has integrated the Yag-suite into the CI/CD chain, enhanced the expertise of the development team, and increased his mastering of the cyber risk carried by the applications.

For the Developers

In just 1 to 2 days of training, the developers took a hands-on the features which enabled them to launch the automatic analysis of their code, obtain a diagnostic of potentially critical vulnerabilities and proceed to their remediation.

Key outcomes

- ✓ More autonomy in mastering the cyber risk carried by the applications
- ✓ Better efficiency in fixing source code vulnerabilities
- ✓ Upgrade of developers' expertise
- ✓ Reduction of onboarding time
- ✓ Gain of serenity in deliveries
- ✓ Time and money savings

Get in touch to see how we can support you in your code securing approach

Let's talk!