Security

Security aspects are a major concern of the OpenEMIS team. Indeed, since the OpenEMIS solution is deployed within government agencies (such as Ministries of Education), data security is an inherent aspect of the implementation of such a project.

While security audits are organized internally on a recurrent basis to check the integrity and security of the OpenEMIS Application stack, some analysis and studies were also produced by external government organizations.

Security aspects are managed at 2 levels: a software level and a network level.

Software Level

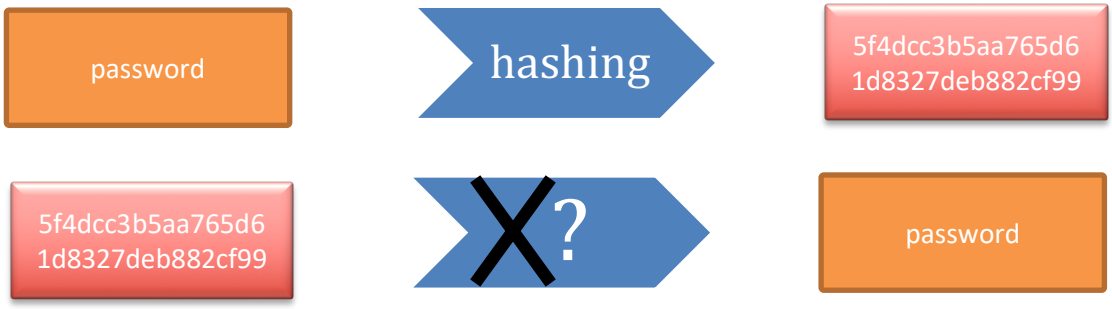
The CakePhp Framework, adopted by OpenEMIS, addresses many security vulnerabilities natively. CakePhp has one of the most active developer communities. One of the major concerns of this community is the security of the CakePhp engine. Its native library comes with several modules working on application security. In addition to native modules, a large number of extensions are offered by the developer community on security aspects.

OpenEMIS is regularly audited to prevent the most known attacks on the web. OpenEMIS follows very closely the recommendations of the OWASP foundation (Open Web Application Security Project) regarding Internet security (web applications).

OWASP is a non-profit project designed to raise awareness about application security by identifying some of the most critical risks faced by businesses. The OWASP project community regularly ranks the 10 most critical vulnerability risks. This ranking is called OWASP Top 10.

Below is a list of some of the attacks tested during these audits and how OpenEMIS apprehends and manages this type of attack:

Password Hashing:



Password hash is one of the most basic security practices that must be performed. Without it, every stored password can be stolen if the storage medium (typically a database) is compromised. This password can then be immediately used to fraudulently access not only one application but also other applications if the user uses the same password elsewhere.

By applying a hash to the password before storing it, you make it very difficult for an attacker to know the original password, and you still have the ability to compare the hashed password to a received string.

OpenEMIS uses the hash functions available in the CakePhp library to set up its own secure password storage logic.

On request we also offer our customers an additional security enhancement mechanism for password management. This is a technique consisting of password hashed an additional data (which we only know). In cryptography this technique is called "A grain of salt, or "salt". It is applied during the hash process to eliminate the possibility of dictionary attacks (hashes recorded in a large list and compared in use an algorithm using the brute force principle).

SQL Injection and Code Injection

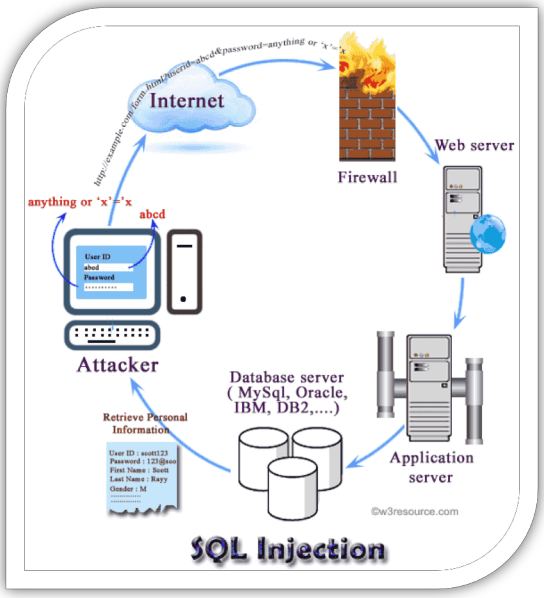
The SQLi flaw, abbreviation of "SQL Injection", is a group of security flaw operating methods of an application interacting with a database. It allows to inject in the SQL query in progress a piece of query not foreseen by the system and which can compromise its security.

An injection flaw, such as SQL, OS and LDAP injection, occurs when unreliable data is sent to an interpreter as part of a command or query. The attacker's hostile data can fool the interpreter into executing accidental commands or accessing unauthorized data. This type of attack is placed first in the "OWASP Top 10" ranking.

Preventing Injection requires separating unreliable data from commands and queries.

The best option is to use a healthy API that avoids any use of the interpreter or provides a customizable interface.

These customizable APIs manage a set of mechanisms and best practices (including escaping special characters using interpreter-specific escape syntax).



The CakePHP ORM provides abstraction for the most commonly used SQL functions. This ORM and database abstraction layers prevent most SQL injection problems. Our teams are constantly on the lookout, enabling us to implement new discoveries in the field of safety.

Brute-force Attack

A brute-force attack, or comprehensive attack, is a method used by hackers and in cryptanalysis to discover the password or key. Exhaustiveness because it is a question of trying all possible combinations.

Several means are implemented by our technical teams to prevent brute force attacks. We can quote among others:

Use of robust passwords
The first defence against this attack is to strengthen the password by avoiding the pitfalls that brute-force optimized attacks exploit. Reinforcing the password consists of :

- Lengthen the password or key if possible (the longer the password, the more difficult it is to hack)
- Use the widest possible range of symbols (lower case, upper case, punctuation, numbers); the introduction of national characters (Â, ÿ...) makes it more difficult for pirates to work (but sometimes also to enter their password when abroad)

OpenEMIS allows the customer to set the minimum password length when creating or updating user accounts. We strongly recommend that our clients choose a number greater than 8.

Random password generation
We implement random password generation algorithms.

Temporal limitation of connections
The CakePhp community has a plugin that manages the number of connection attempts. This is a tool for setting the maximum number of unsuccessful login attempts and the time to wait after an account is blocked. We use this plugin during the authentication process.

Falsification of inter-site CSRF requests
A Cross Site Request Forgery (CSRF) attack forces an authenticated victim's browser to send a forged HTTP request to a vulnerable web application, including the victim's session cookie and any other information automatically included.

This allows the attacker to force the victim's browser to generate requests that the vulnerable application believes are legitimate requests from the victim. This attack ranks 8th in the "OWASP Top 10" ranking.

Prevention:

- Request user confirmations for critical actions.
- Use of validity tokens in forms: This is to ensure that a mailed form is accepted only if it has been produced a few minutes previously: the validity token will be proof of this. The validity token must be transmitted as a parameter and checked on the server side. The Cakephp CSRF

Component we use provides protection against CSRF attacks. This component is provided in the Cakephp Core.

- Avoid using HTTP GET requests to perform actions: this technique will naturally eliminate simple attacks based on images, but will let pass attacks based on JavaScript, which are very simply capable of launching HTTP POST requests.
- Checking the referrer in sensitive pages: knowing the client's source makes it possible to secure this type of attack. This consists in blocking the client's request if the value of its referrer is different from the page from which it should theoretically come.

Cross Site Scripting (XSS)

XSS (more officially called Cross-Site Scripting) is a flaw that allows HTML or JavaScript code to be injected into poorly protected variables. This is the third attack of the "OWASP Top 10".
XSS vulnerabilities occur whenever an application accepts unreliable data and sends it to a web browser without proper validation.

XSS allows attackers to execute scripting in the victim's browser to hijack user sessions, disfigure websites, or redirect the user to malicious sites.

There are actually two types of XSS:

- Reflected XSS (not permanent)
    - This rift is the simpler of the two. It is called non-permanent because it is not stored in a file or database. So it is ephemeral.
- The XSS stored (permanent)
    - The permanent flaw is the most serious XSS flaw because the script is saved in a file or a database. It will therefore be displayed each time the site is opened.

Prevention:
As this attack is not managed natively by the CakePHP framework, we implemented our own defense mechanism to counter this type of attack.

The most suitable solution against this flaw consists in the validation of any information sent by users to the system. It is a question of using good development practices allowing the technical and business validation of the data. The use of character chain sanitization functions occupies a prominent place in the prevention of this type of attack.

Php's *htmlspecialchars()* function filters symbols of type <, & or ", replacing them with their HTML equivalent. The CakePhp API offers several methods allowing us to sanitize the character string before saving it to the database.

Denial of Service (DoS)

A denial of service attack (Denial of Service attack, hence the abbreviation DoS) is a computer attack designed to render a service unavailable, to prevent legitimate users of a service from using it. It can be a question of:

- flooding of a network in order to prevent its operation;
- disruption of connections between two machines, preventing access to a particular service;
- obstructing access to a service to a particular person;

- also sending billions of bytes to an internet box.

Denial of service attacks can block a file server, make it impossible to access a web server or prevent email distribution in a company.

A known flaw in the PHP Framework that we use allowed attacks of this type. The CakePHP community reacted quickly by offering a patch that we implemented on our platform.

Code injection
See points about SQL Injection and XSS

File Recovery
One of the most critical issues for applications is securing data, including documents hosted on a data server. This vulnerability is better known as an unsecured direct reference to an object. A direct reference exists when a developer displays a reference to the internal implementation of an object, such as a file, a directory, a database record, a key such as a parameter of a form or URL. An attacker can then manipulate the object reference directly to access other objects without authorization, even with access control checks in place.

Protection:
The best protection is to avoid exposing a direct reference to an object to the user, by using an index, an indirect reference equivalence or another indirect method that is easy to validate. If a direct reference to an object is to be used, we verify that the user is authorized before using it.

The implementation of a reference method to the application objects is important:

- Avoid exposing private object references to users, whenever possible, such as primary keys or file names
- Validate all references to private objects without reserve, via the method of acceptance of good values
- Check authorization to all referenced objects

Phone Home Functions

Phoning Home (also called Phone Home or Call Home), in computing, refers to an act of client to server communication which may be undesirable to the user and/or proprietor of the device or software. It is often used to refer to the behavior of security systems which report network location, username, or other such data to another computer. The OpenEMIS Source code doesn't contain any legal or malicious phone home function.

Backdoor Functions

A backdoor is a method, often secret, of bypassing normal authentication or encryption in a computer system, a product, or an embedded device. Backdoors are often used for securing remote access to a computer, or obtaining access to plaintext in cryptographic systems. The OpenEMIS Source code doesn't contain any backdoor function.

Network Level

1. Security - Firewalls:

A self-hosted solution should include 3 groups of firewalls:

- The first group is intended to protect the demilitarized zone (web servers) by firewall equipment.
- The second group is intended to provide an additional level of protection for internal local virtual servers, in particular those hosting databases. These are software firewalls.
- The third group is the WAF (Web Application Firewall): To protect applications from external threats and ensure their continuity of service at all times, an application firewall is used.
- The WAF meets the needs of:
    - Availability of internal and external websites
    - Ensure a high level of protection for all information system applications (OWASP TOP 10 attack blocking such as: SQL Injection - brute force - file recovery - code injection - cross site scripting (XSS) - cross site request forgery (CSRF) - denial of service (DoS) etc)
    - Accelerate application performance with Transparent Reverse Proxy
    - Protection of sensitive data
    - Securing Business Web Applications
    - Security of third-party applications used by your company (i.e. SharePoint, OWA, etc.)
    - Mobile Application Security
    - Reduce the risk of identity theft and access to illegitimate features or data

All firewalls include a logging module and an IPS module, to quickly identify potential attacks. To further increase web and application security, all servers use only private IP addresses and none can be accessed directly from the Internet.

2. Security

Communications: Encryption of exchanges via the HTTPS protocol (SSL Certificate).