A Sluitappel 67
5491 TS St-Oedenrode
Nederland

T +31 (0)413 820285
E info@fruitcake.nl
W www.fruitcake.nl

IBAN NL51 RABO 0130 2626 68
BTW NL8174 55 188 B01
KvK 17200101

Fruitcake Security Program & Software

Anti-virus / malware policy

Connected hardware is checked on a regular basis for anti-virus and malware. Anti-virus software is installed to ensure a safe as possible working environment for the team members.

Sensitive information

Customer data is stored on production servers and only available for employees that are given access. Back-ups are stored on restricted environments , depending on the platform.

Sensitive data will be stored encrypted upon request by the client.

Documentation and data provided by the client will be stored on Google Drive or Google Mail. Our Google accounts are protected with 2- Factor.

Clients can request data to be deleted. Sensitive information on paper will be destroyed by shredding the information.

Data access

Only employees working on the product will have access to production environments. Data access is revoked upon termination.

Technology

Used technology

Different technologies are used, but most are based on PHP and MySQL, for example:

- Laravel; see below
- Drupal; see https://www.drupal.org
- Magento; see http://devdocs.magento.com/guides/v2.2/architecture/security_intro.html

Security updates will be applied when a security contract is agreed upon with the Client. Otherwise update can be requested or can be done on hourly-based contracts.

A Sluitappel 67
5491 TS St-Oedenrode
Nederland

T +31 (0)413 820285
E info@fruitcake.nl
W www.fruitcake.nl

IBAN NL51 RABO 0130 2626 68
BTW NL8174 55 188 B01
KvK 17200101

Hosting provider

Different hosting options are available, with different techniques and security. The most common are:

- Flexwebhosting : Shared hosting, password-based restrictions
- Digital Ocean/AWS/Linode hosting with Laravel Forge: SSH key access
- Custom: Dependent on the hosting provider

Laravel Forge, Digital Ocean, AWS, Linode and our FXW dashboards are protected with 2-factor. Only required employees are given SSH access using SSH keys to the servers.

For Laravel projects, a local development server (located physically inside the office) will be used. This will prevent unauthorized access or data loss from any possible break-in. The local development server is operated within the Fruitcake office. The office is protected with an alarm system, personal entry for Fruitcake team members by key. The local development server is situated within a closed section of the office.

Sensitive data will only be available in the online test- and production environment, not on the (local) development environment.

Security components for Laravel projects

Secure communication: HTTPS

We will request a SSL certificate to provide a secure connection between the server and the client.

SSH Access: Private keys

SSH connections to the production servers will be available through private keys, secured with a passphrase. The private keys will be given out only to a limited group of developers who really need it.

Password hashing: Bcrypt with unique salt

All user passwords will be hashed one-way, with an unique salt per password. These passwords cannot be reversed and because of the salt, rainbow tables are useless. The passwords will never be stored in plain-text.

fruitcake.

A Sluitappel 67
5491 TS St-Oedenrode
Nederland

T +31 (0)413 820285
E info@fruitcake.nl
W www.fruitcake.nl

IBAN NL51 RABO 0130 2626 68
BTW NL8174 55 188 B01
KvK 17200101

Encryption of sensitive data: AES-256-CBC

If needed, sensitive data can be encrypted using AES encryption (AES-256-CBC cipher), with an encryption key. This key is generated on the production server and stored on file, outside the webfoot and outside of version management.

SQL Injection: Parameterized queries

To prevent SQL injection, an ORM (Eloquent) is used, which only uses parameterised queries. This way, it is never possible to inject malicious user-supplied input into queries.

XSS Protection: Safe output by default

To prevent XSS, a secure tempting engine (Twig or Blade) is used. Variable escaping is turned on by default, so only when it is explicitly required and output is trusted, it will have raw output.

CSRF Protection: On by default

A CSRF token is generated on every session and required on all POST/PUT/DELETE requests by default. This will require that every form uses the CSRF token. This way, cross site requests are not possible.

Authentication and Authorization

Authentication is built-in in Laravel, which uses secure practices as described above (secure cookies, hashed passwords etc). It also provides mechanisms to protect routes based on the login status and can be used to verify certain roles for certain areas.

Validation

Laravel includes a validation component, to securely validate the user input, for example required files, expected types (images/numbers/emails) or unique/existing rows.

Configuration variables

Sensitive configuration is saved in a '.env' file, which is not stored in the GIT repository and not in the webroot, but only accessible on the production server. This will be used to store database passwords, encryption keys and key/secrets for services like AWS.

GIT versioning and backups

The application source will be stored in a git repository. This repository will not contain any sensitive information and/or database passwords