Site Security in Drupal 8

rlhawk

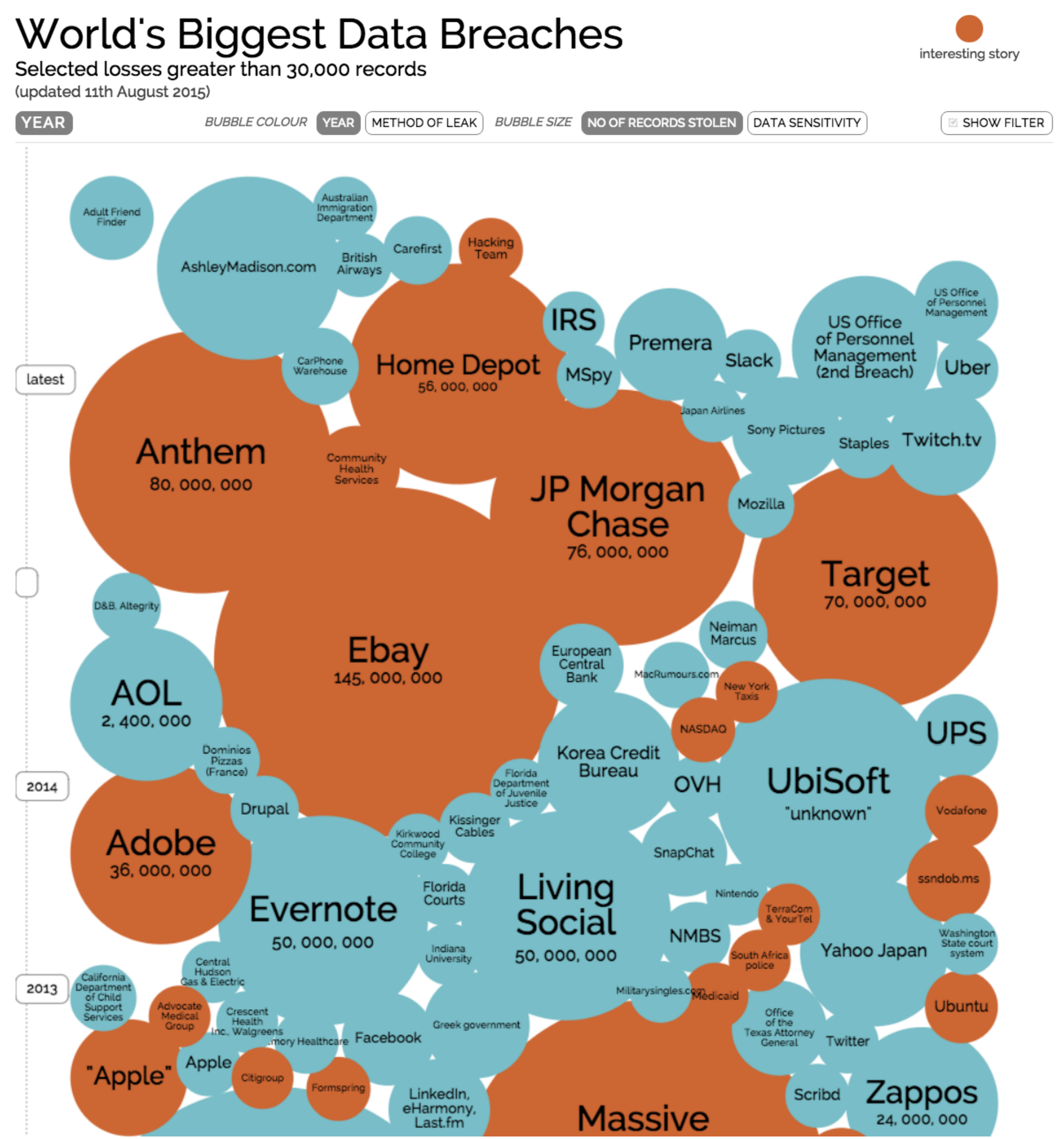
- Web developer, designer, and consultant
- Organizer of Seattle Drupal Users Group
- Module maintainer
  - Encrypt
  - Key
  - Encrypt Form API
  - Encrypted Files
  - Encrypt User
  - Townsend Security Key Connection
- Fan of keeping my stuff safe

How do you feel when you think about web security?

- Overwhelmed
- Confused
- Anxious
- Hopeless
- Paranoid
- In denial

Sticking your head in the sand and hoping your site doesn’t get compromised is not a viable security strategy.

- Empowered
- Informed
- Comforted
- Confident

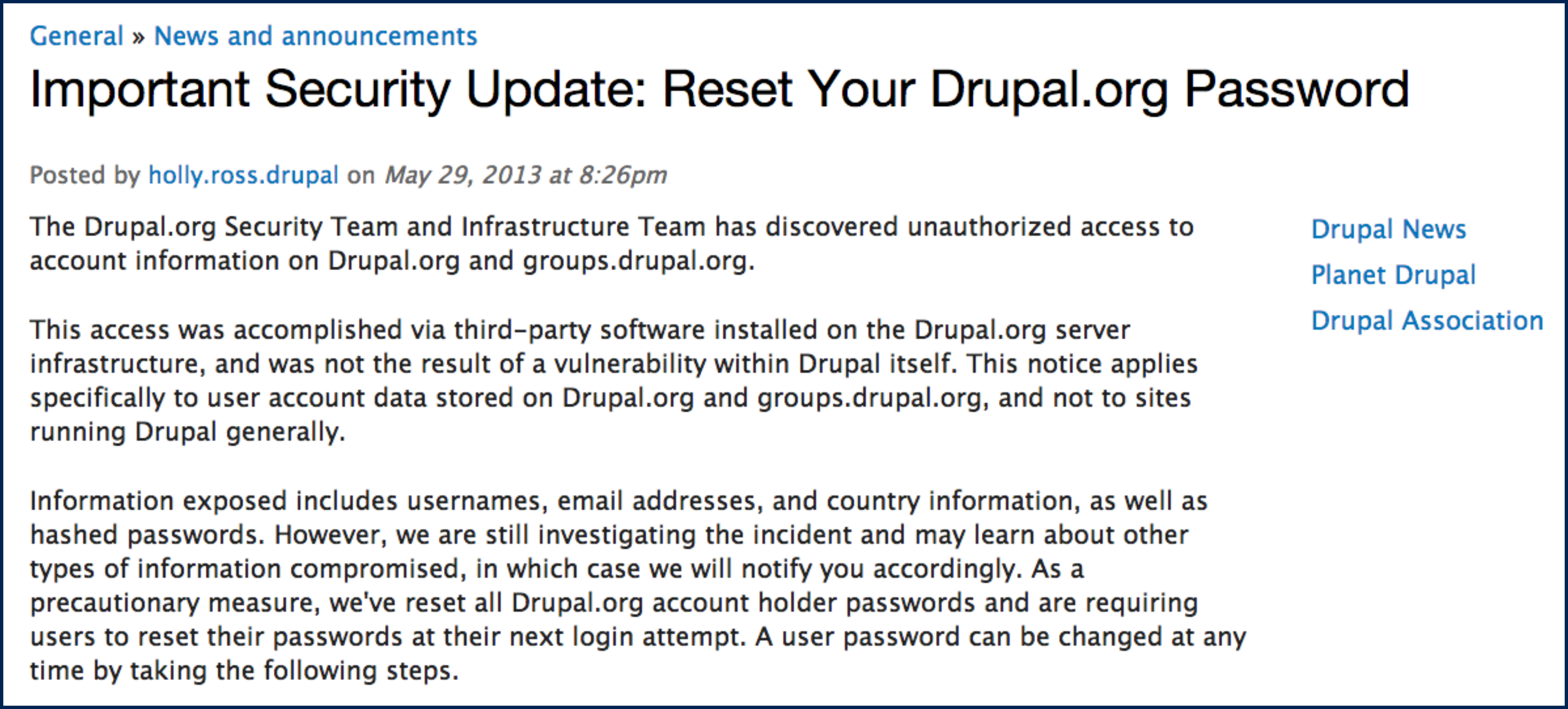


World's Biggest Data Breaches

Selected losses greater than 30,000 records

(updated 11th August 2015)

Source: http://tiny.cc/d8-security-01



General » News and announcements

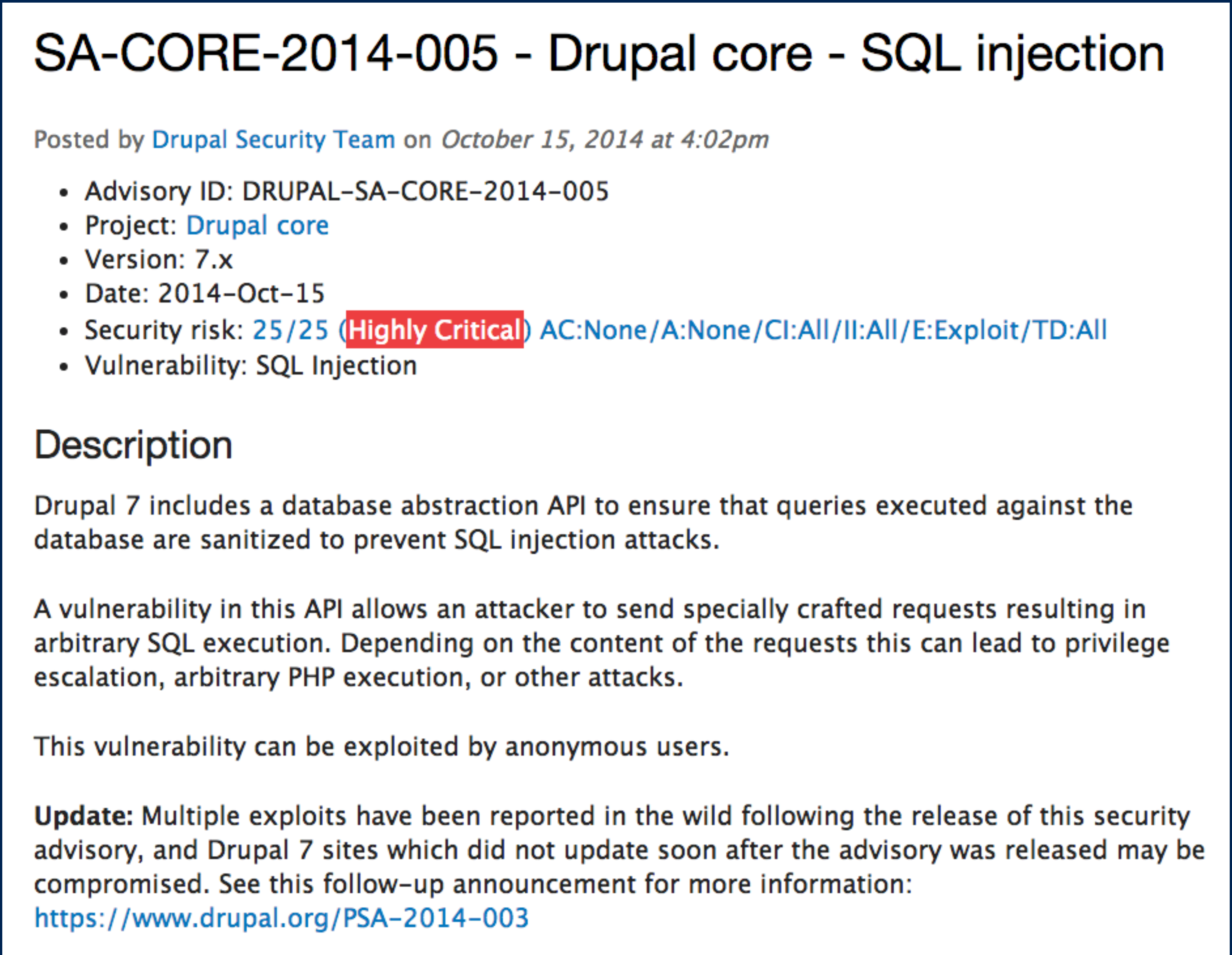
Important Security Update: Reset Your Drupal.org Password

Posted by holly.ross.drupal on *May 29, 2013 at 8:26pm*

The Drupal.org Security Team and Infrastructure Team has discovered unauthorized access to account information on Drupal.org and groups.drupal.org.

This access was accomplished via third-party software installed on the Drupal.org server infrastructure, and was not the result of a vulnerability within Drupal itself. This notice applies specifically to user account data stored on Drupal.org and groups.drupal.org, and not to sites running Drupal generally.

Information exposed includes usernames, email addresses, and country information, as well as hashed passwords. However, we are still investigating the incident and may learn about other types of information compromised, in which case we will notify you accordingly. As a precautionary measure, we've reset all Drupal.org account holder passwords and are requiring users to reset their passwords at their next login attempt. A user password can be changed at any time by taking the following steps.



SA-CORE-2014-005 - Drupal core - SQL injection

Posted by Drupal Security Team on *October 15, 2014 at 4:02pm*

- Advisory ID: DRUPAL-SA-CORE-2014-005
- Project: Drupal core
- Version: 7.x
- Date: 2014-Oct-15
- Security risk: 25/25 (Highly Critical) AC:None/A:None/CI:All/II:All/E:Exploit/TD:All
- Vulnerability: SQL Injection

Description

Drupal 7 includes a database abstraction API to ensure that queries executed against the database are sanitized to prevent SQL injection attacks.

A vulnerability in this API allows an attacker to send specially crafted requests resulting in arbitrary SQL execution. Depending on the content of the requests this can lead to privilege escalation, arbitrary PHP execution, or other attacks.

This vulnerability can be exploited by anonymous users.

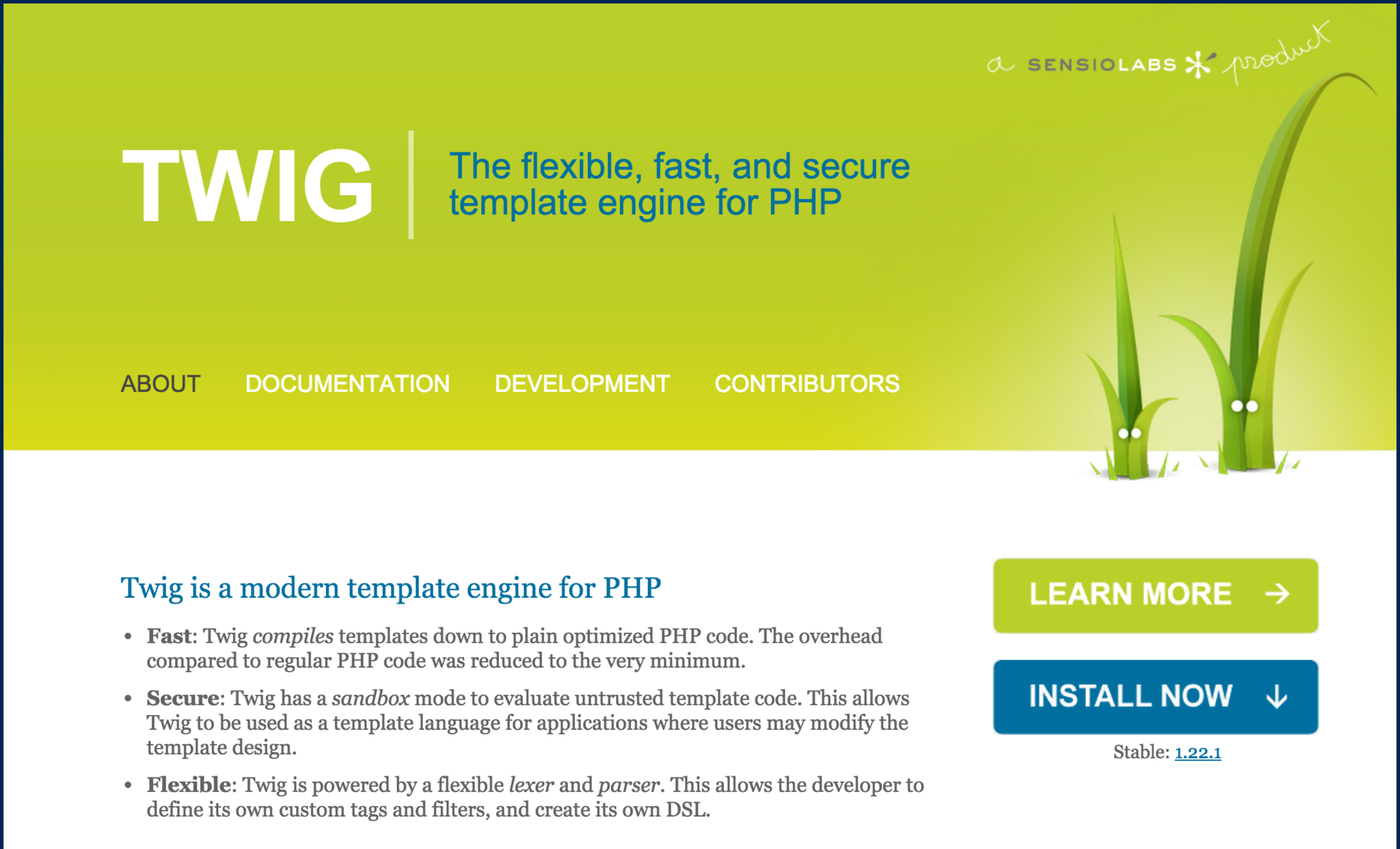
Update: Multiple exploits have been reported in the wild following the release of this security advisory, and Drupal 7 sites which did not update soon after the advisory was released may be compromised. See this follow-up announcement for more information: https://www.drupal.org/PSA-2014-003

Security release windows

- Every Wednesday for contributed projects
- One Wednesday a month (usually the third Wednesday) for Drupal core

Security improvements in Drupal 8

New default theme system

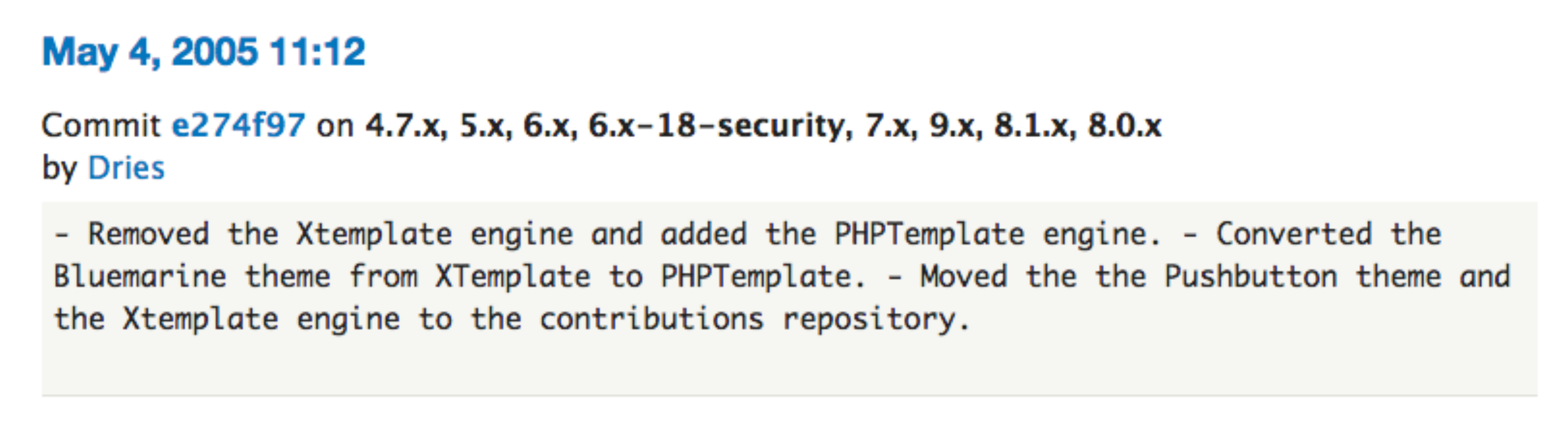


TWIG | The flexible, fast, and secure template engine for PHP

Twig is a modern template engine for PHP

- **Fast**: Twig *compiles* templates down to plain optimized PHP code. The overhead compared to regular PHP code was reduced to the very minimum.
- **Secure**: Twig has a *sandbox* mode to evaluate untrusted template code. This allows Twig to be used as a template language for applications where users may modify the template design.
- **Flexible**: Twig is powered by a flexible *lexer* and *parser*. This allows the developer to define its own custom tags and filters, and create its own DSL.

Stable: 1.22.1



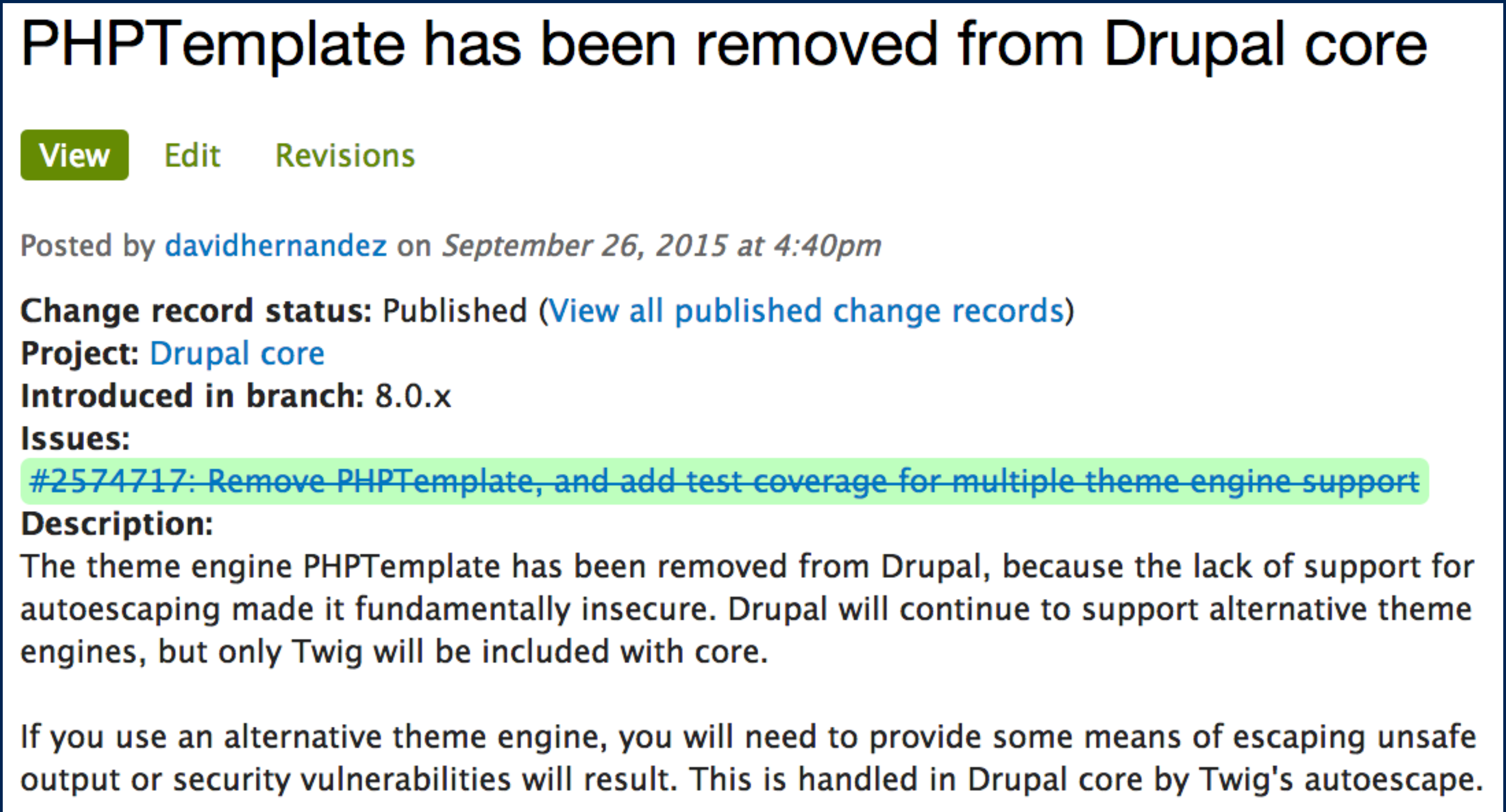
May 4, 2005 11:12

Commit e274f97 on **4.7.x, 5.x, 6.x, 6.x-18-security, 7.x, 9.x, 8.1.x, 8.0.x**
by Dries

```
- Removed the Xtemplate engine and added the PHPTemplate engine. - Converted the
Bluemarine theme from XTemplate to PHPTemplate. - Moved the the Pushbutton theme and
the Xtemplate engine to the contributions repository.
```

PHPTemplate

- Encouraged bad practices by allowing any PHP code to be executed in a template file
- Doesn’t encourage separation of logic and presentation code
- In Drupal 8, modules rely on automatic escaping at the theme layer when outputting variables
- PHPTemplate can’t automatically escape variables



PHPTemplate has been removed from Drupal core

Posted by davidhernandez on *September 26, 2015 at 4:40pm*

Change record status: Published (View all published change records)
Project: Drupal core
Introduced in branch: 8.0.x
Issues:
~~#2574717: Remove PHPTemplate, and add test coverage for multiple theme engine support~~
Description:
The theme engine PHPTemplate has been removed from Drupal, because the lack of support for autoescaping made it fundamentally insecure. Drupal will continue to support alternative theme engines, but only Twig will be included with core.

If you use an alternative theme engine, you will need to provide some means of escaping unsafe output or security vulnerabilities will result. This is handled in Drupal core by Twig's autoescape.

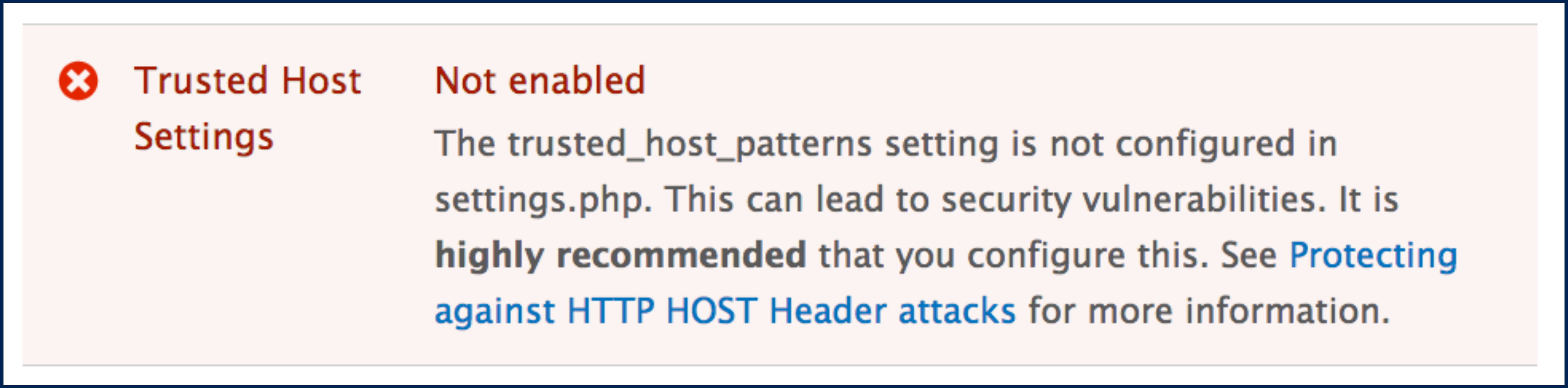
Twig security

- Limits what actions can be performed in a template file
- Provides automatic escaping of template variables
- More information: https://www.drupal.org/theme-guide/8/twig

PHP Filter module removed from core

- Added a text filter (“PHP Evaluator”)
- Added a text format (“PHP Code”)
- Allowed execution of arbitrary PHP in a field that supports text formats (Body, for instance)
- One of the tools that attackers can employ once they have access to a Drupal site’s database
- Removing it encourages better development practices

Trusted host patterns



☒ Trusted Host Settings

Not enabled

The trusted_host_patterns setting is not configured in settings.php. This can lead to security vulnerabilities. It is **highly recommended** that you configure this. See Protecting against HTTP HOST Header attacks for more information.

Dynamic base URL detection

- Drupal determines what it considers to be the base URL of the site dynamically, if the base URL is not explicitly set

Concerns with dynamic base URL detection

- Emails sent with links to another domain
- Cache entries can contain the wrong domain
- More info: https://www.drupal.org/node/1992030

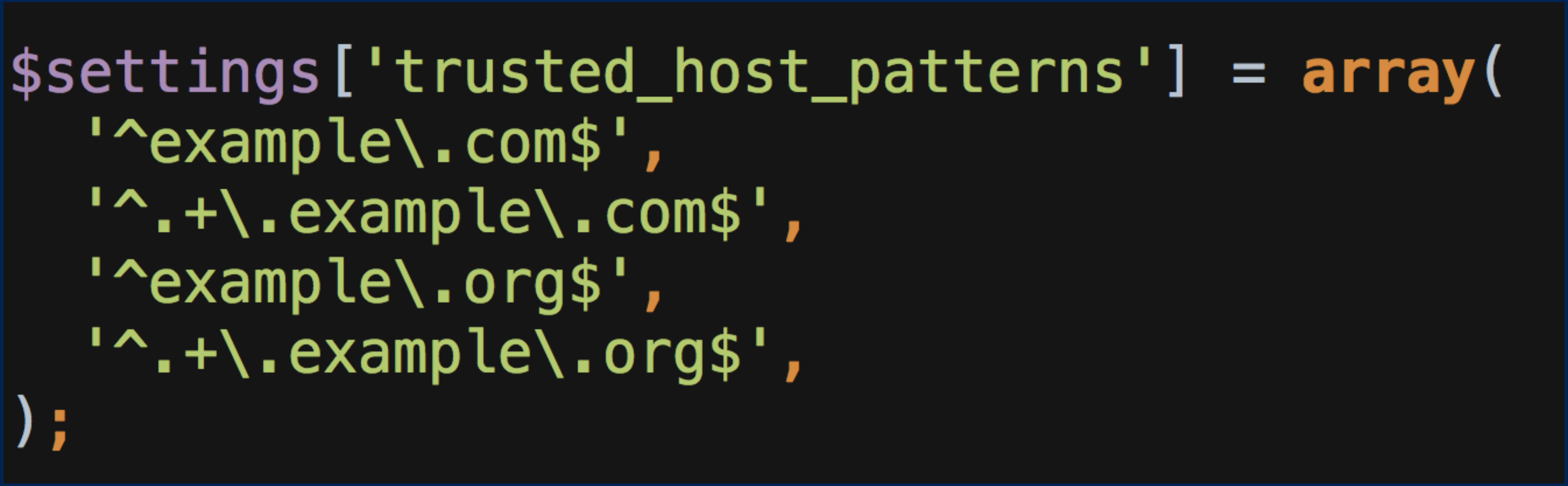
Solutions

- Change your web server configuration

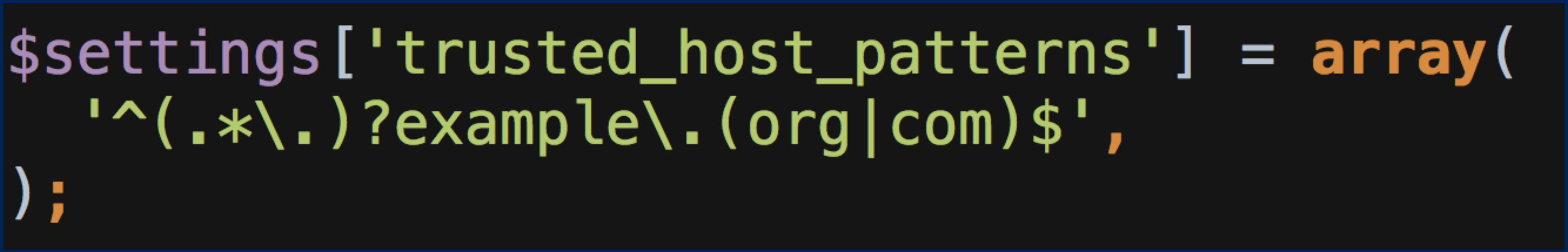
- Drupal 7: Set a specific domain by defining $base_url in settings.php

- Drupal 8: Define trusted host patterns in settings.php

Trusted host patterns



```
$settings['trusted_host_patterns'] = array(
  '^example\.com$',
  '^.+\.example\.com$',
  '^example\.org$',
  '^.+\.example\.org$',
);
```



```
$settings['trusted_host_patterns'] = array(
  '^(.*\.)?example\.(org|com)$',
);
```

Session IDs

- Session IDs uniquely identify a visitor to the site
- Session IDs are stored in a cookie in a user's browser
- Session data is stored in the database
- The session ID is sent by the browser to the server on every page request

Session IDs in the database

- In Drupal 7, the session ID is stored in plaintext
- In Drupal 8, the session ID is hashed

Hashed session IDs

- Prevents session hijacking if session data in the database is compromised

- Details for the curious: Uses SHA-256, Base64-encoded, and made URL-friendly

Passwords are “stretched” more

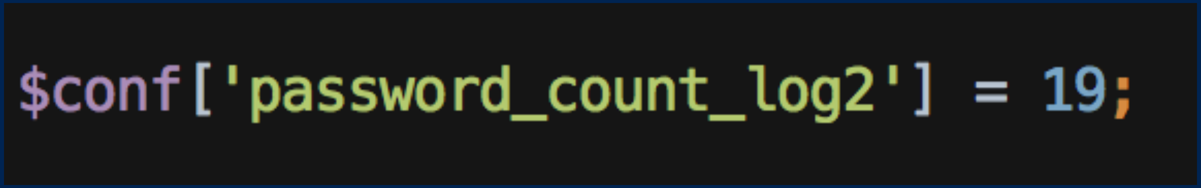
- Password stretching is a way of making brute-force attacks more computationally expensive
- Passwords are run through the hashing algorithm many times to slow down the process

Log2 number increased by 1 in Drupal 8

- Accounts for increases in computing power over time
- The password is run through the hashing algorithm 2^x times
- In Drupal 7, the log2 number is 15 (so the password is run through the hashing algorithm 32,768 times)
- In Drupal 8, the log2 number is 16 (so 65,536 times through the algorithm)

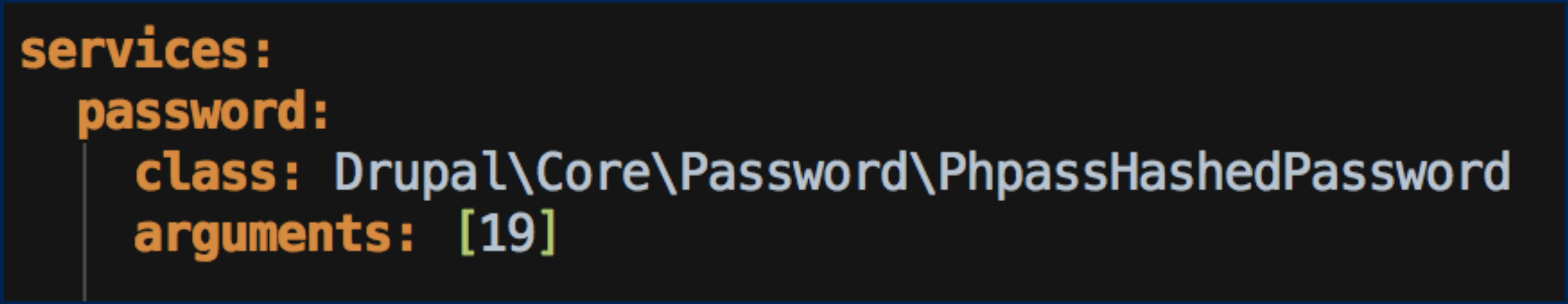
The number of iterations can be customized

- In Drupal 7, add this to settings.php:



```php
$conf['password_count_log2'] = 19;
```

- In Drupal 8, add this to services.yaml:



```yaml
services:
  password:
    class: Drupal\Core\Password\PhpassHashedPassword
    arguments: [19]
```

The number of iterations can be customized

- Minimum log2 hash count is 7
- Maximum log2 hash count is 30
- If you use the maximum log2 hash count, the password is run through the hash algorithm 1,073,741,824 times!
- More info: https://www.drupal.org/node/1850524

Database queries are limited to a single statement

- By default, Drupal 8 sets a flag that limits PHP to sending only a single SQL statement at a time
- Works with all database drivers
- It’s possible to override the default in the rare instance that it’s necessary
- Would have greatly reduced the severity of the Drupalgeddon vulnerability

Improved WYSIWG-Text Format Integration

- WYSIWYG editor configuration is integrated with the corresponding text filter

- Adding a button to the WYSIWYG toolbar, automatically adds the appropriate tag to the list of allowed tags

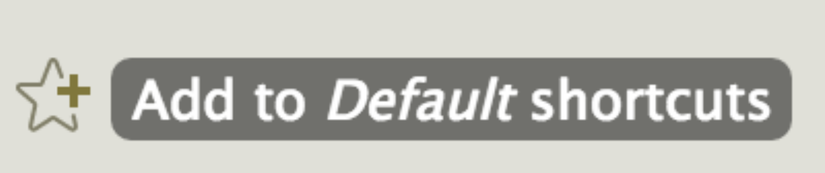
Mixed-mode SSL support removed

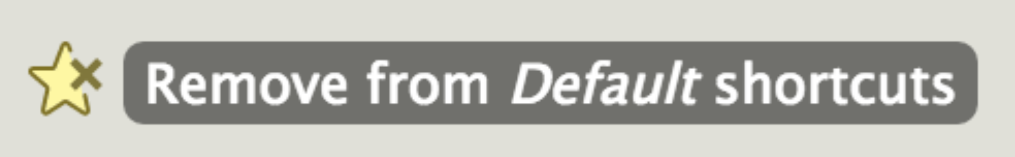
- Was added to Drupal core in order to allow the functionality that the Secure Pages provides
- Allow sites to force some pages to be served over HTTPS, but not others
- Mixed-mode introduced many problems and challenges
- If you really want this functionality, you will have to replace the session handling service, which is now possible in Drupal 8
- Better to enforce HTTPS for all pages at the web server level

Automated CSRF token protection in route definitions

- Makes it easy to add protection to GET requests that cause destructive action or change to configuration

- Examples in Drupal core: when adding or removing a shortcut





- Add in Drupal 8 with: _csrf_token: 'TRUE'

Logout link not protected with a CSRF token

- One link in core that is not protected by a CSRF token is the link to log out (user/logout)
- The Drupal Security Team has determined that this not a serious vulnerability

Security bug bounty program

- Announced in June 2015

- Uses funds from the D8 Accelerate program to offer rewards to users who report security bugs in Drupal 8

- Pays between $50 and $1000 per issue, depending on the seriousness of the bug

- More info: https://www.drupal.org/drupal8-security-bounty

Strategies for improving site security

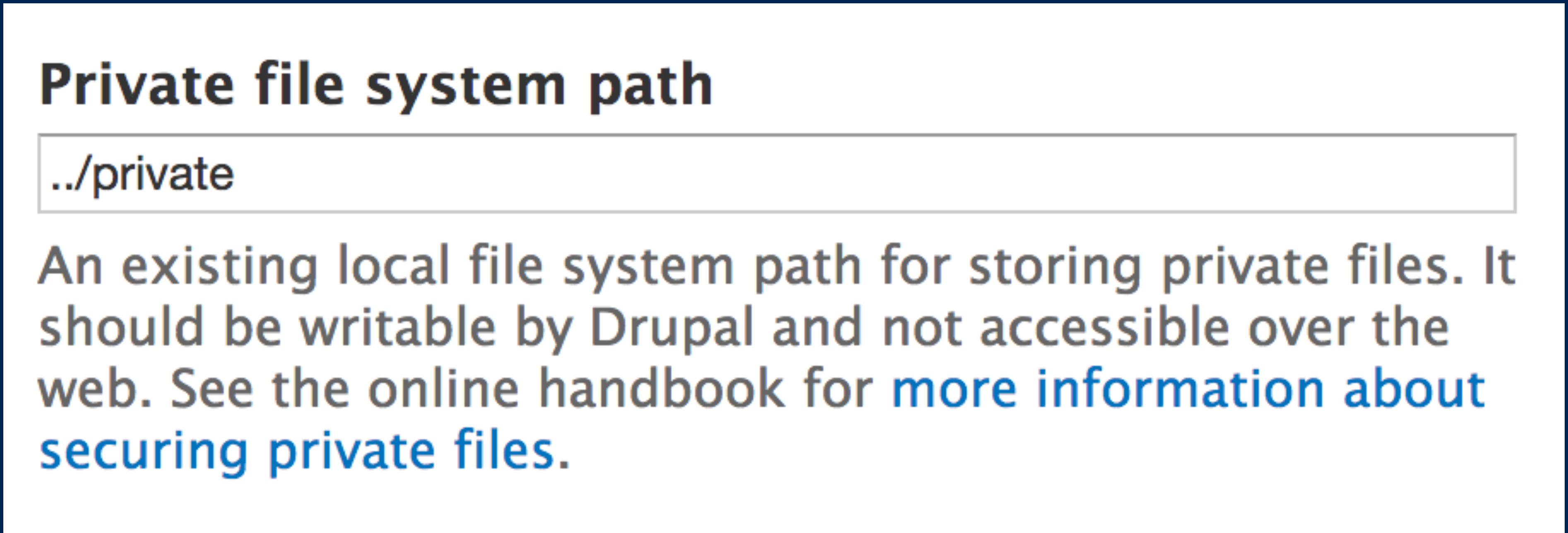
Be careful with roles and permissions in Drupal

- Use the principle of least privilege
- Create a test user to check permissions manually
- Run tests to check permissions
- Use the Paranoia module (D7), which prevents granting risky permissions

Paranoia module

- Disables creation of input formats that use the PHP filter
- Disables editing the User 1 account unless you're logged in as User 1
- Prevents granting risky permissions
- Disables disabling itself
- No Drupal 8 version in development yet, as far as I know.

Keep your private file directory outside of the web root



Private file system path

../private

An existing local file system path for storing private files. It should be writable by Drupal and not accessible over the web. See the online handbook for more information about securing private files.

Use secure file permissions

- User that the web server runs under should only be able to write to public files directory, private files directory, and temporary files directory

Restrict PHP files that can be executed

(Drupal 7 version)

- index.php
- update.php
- cron.php
- authorize.php
- xmlrpc.php

Require HTTPS

- Use a valid SSL certificate
- Prevent data from being exposed
- Prevent session hijacking
- Get a slight search ranking boost from Google

CloudFlare

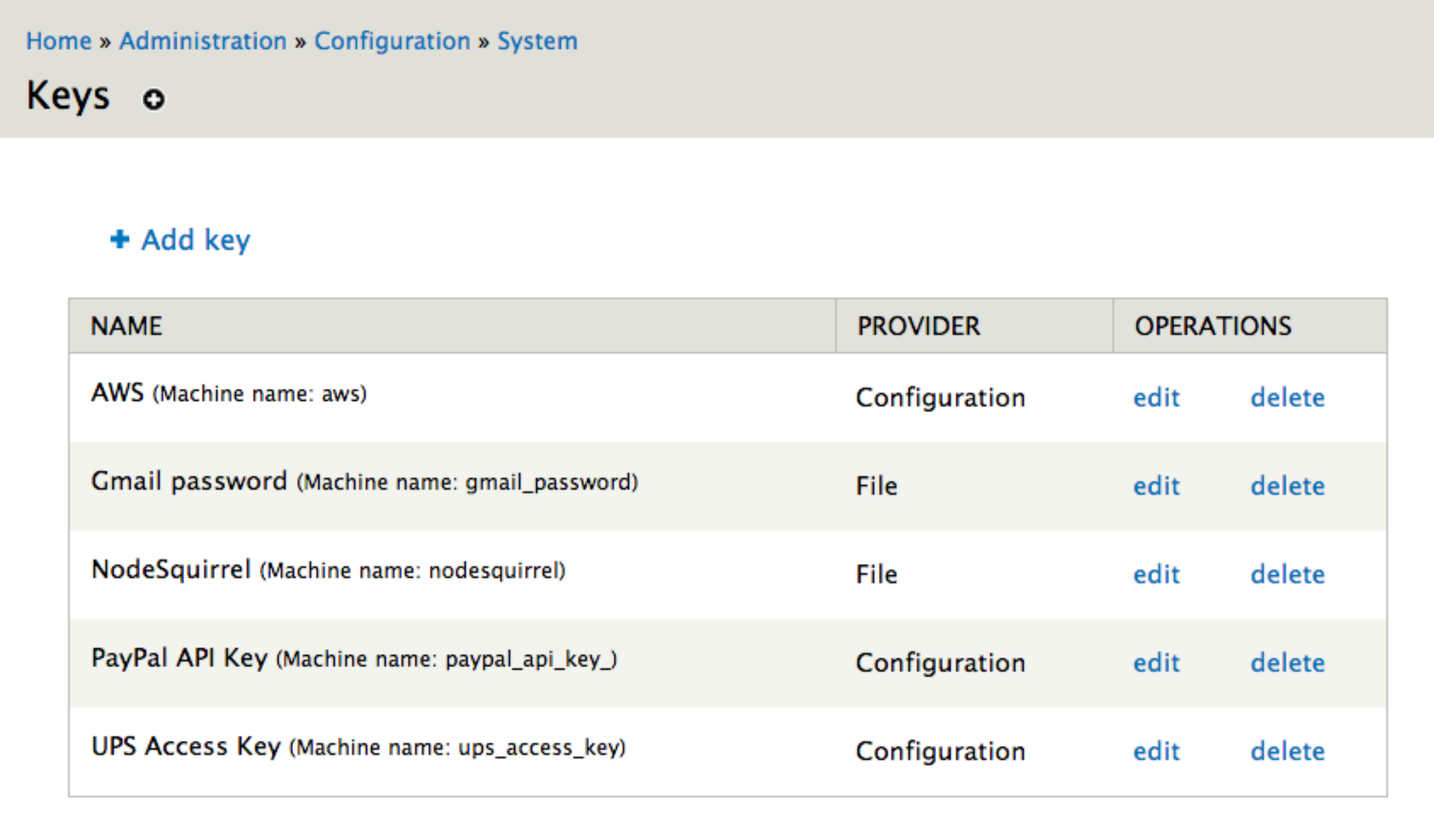
- Route your web traffic through their network
- Free SSL that works with modern browsers
- Many other benefits, some of which require a paid account
- Install the CloudFlare module (optional)

Set up SSL

- Enable SSL support in your web server
- Create a self-signed certificate https://www.digitalocean.com/community/tutorials/how-to-create-a-ssl-certificate-on-apache-for-ubuntu-14-04
- Update your web server configuration to use the certificate
- Enable “always use HTTPS” in CloudFlare
- Test your SSL setup at https://www.ssllabs.com/ssltest/

Key module

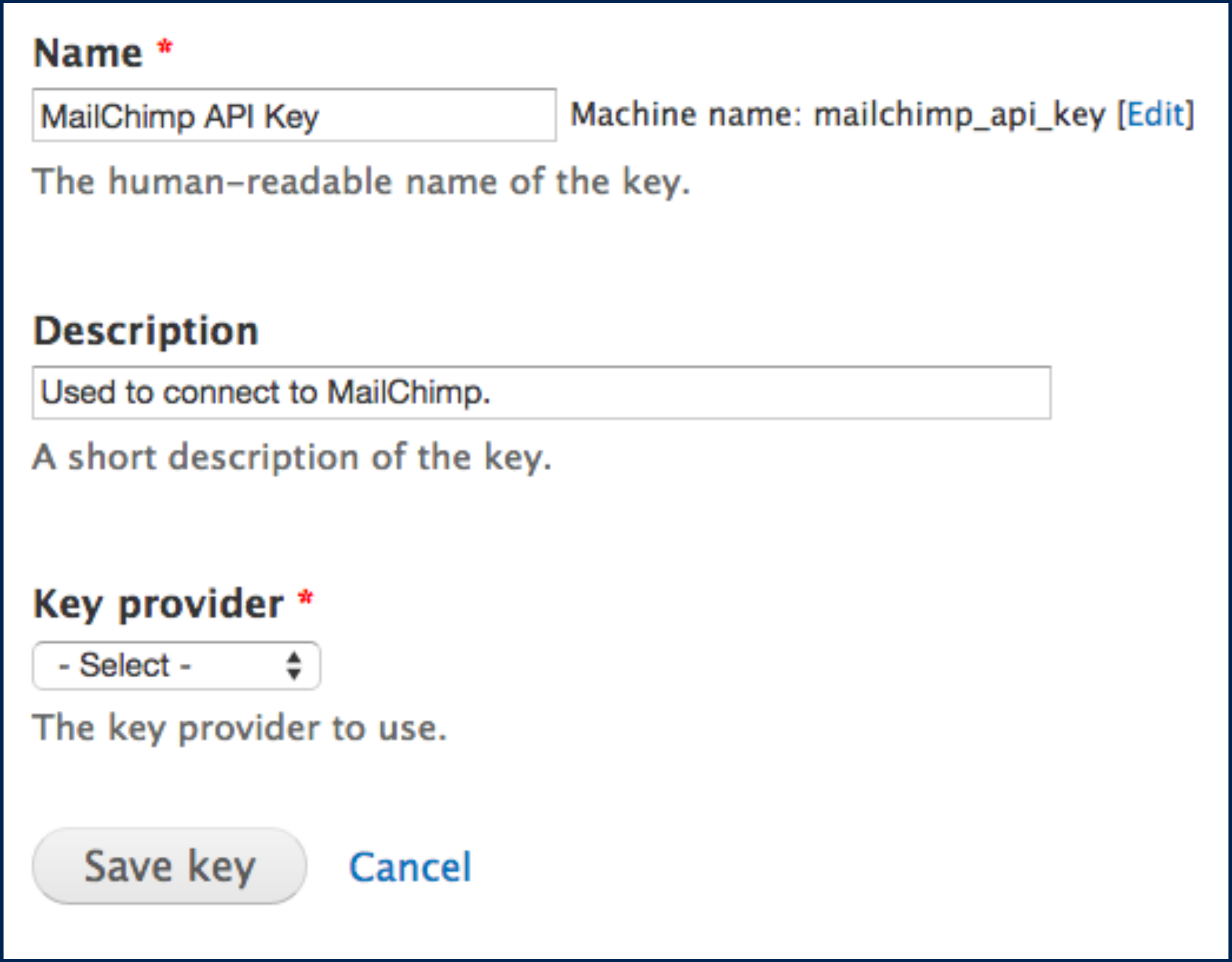
- Manage keys in a single location
- Choose how and where keys are stored (provider)
- Use them when connecting to an external API
- Development for D8 is happening in GitHub: https://github.com/d8-contrib-modules/key



Home » Administration » Configuration » System

Keys

NAME	PROVIDER	OPERATIONS	
AWS (Machine name: aws)	Configuration	edit	delete
Gmail password (Machine name: gmail_password)	File	edit	delete
NodeSquirrel (Machine name: nodesquirrel)	File	edit	delete
PayPal API Key (Machine name: paypal_api_key_)	Configuration	edit	delete
UPS Access Key (Machine name: ups_access_key)	Configuration	edit	delete



Name *

MailChimp API Key Machine name: mailchimp_api_key [Edit]

The human-readable name of the key.

Description

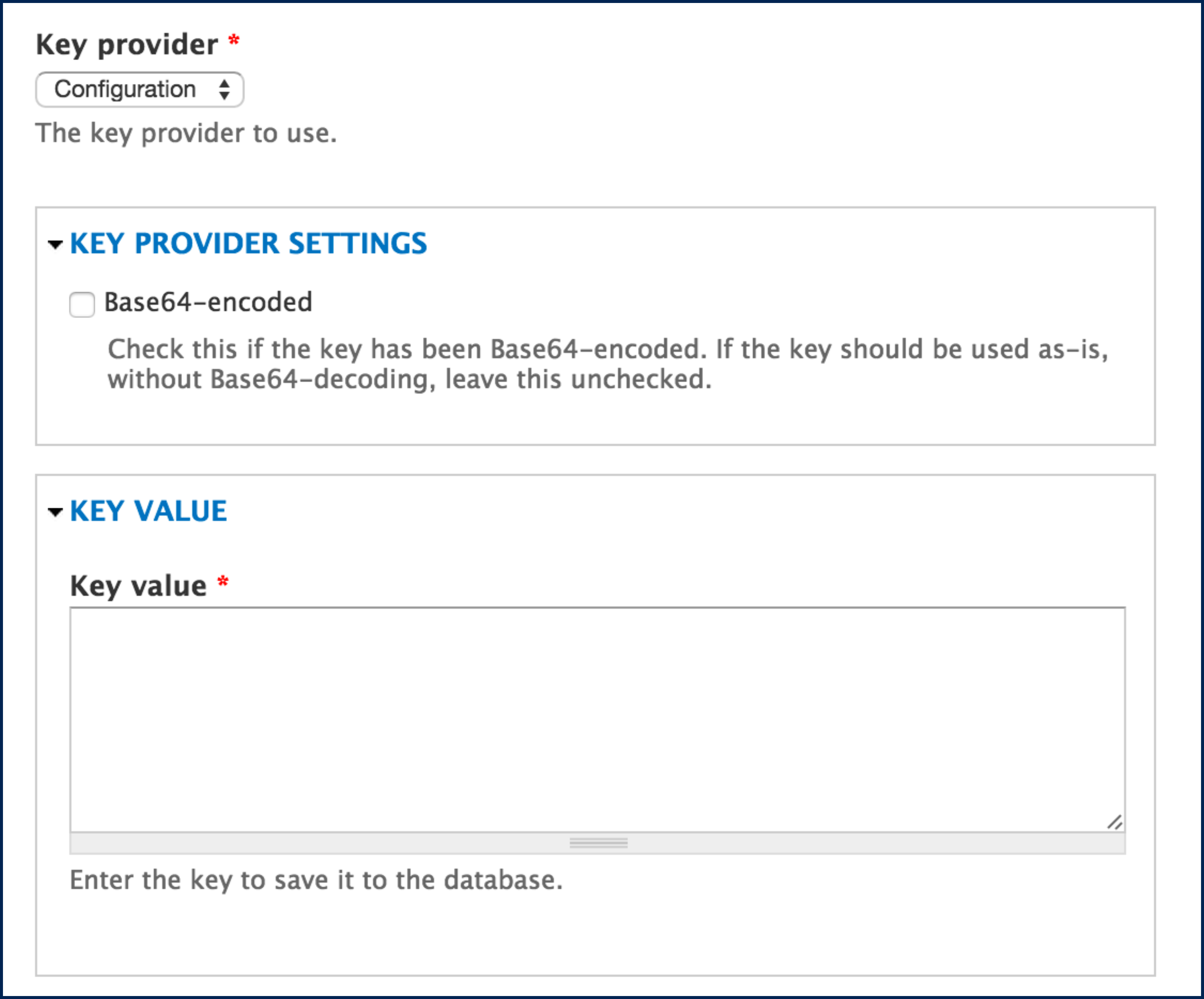
Used to connect to MailChimp.

A short description of the key.

Key provider *

- Select -

The key provider to use.



Key provider *

Configuration

The key provider to use.

▾ KEY PROVIDER SETTINGS

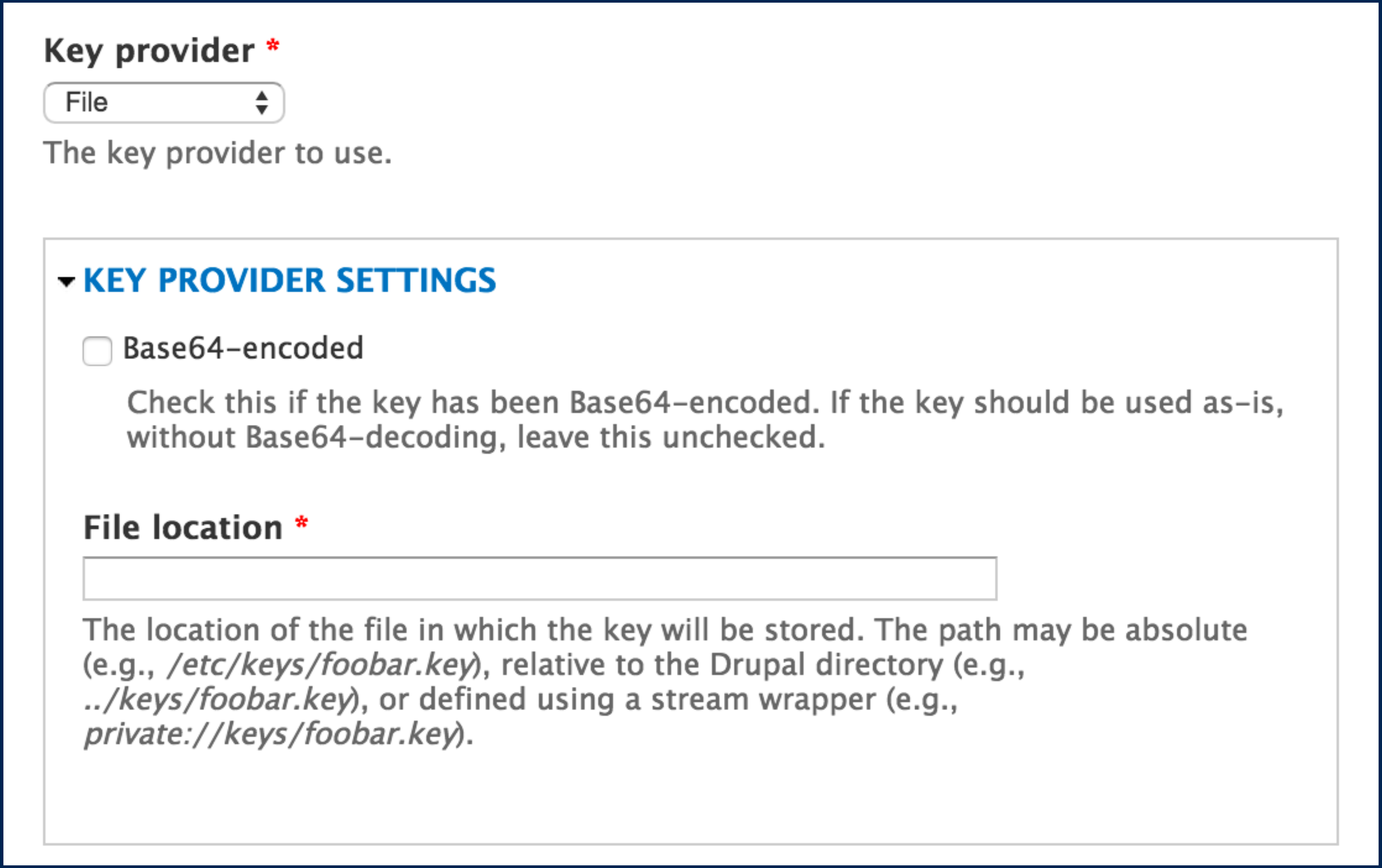
□ Base64-encoded

Check this if the key has been Base64-encoded. If the key should be used as-is, without Base64-decoding, leave this unchecked.

▾ KEY VALUE

Key value *

Enter the key to save it to the database.



Key provider *

File

The key provider to use.

▾ KEY PROVIDER SETTINGS

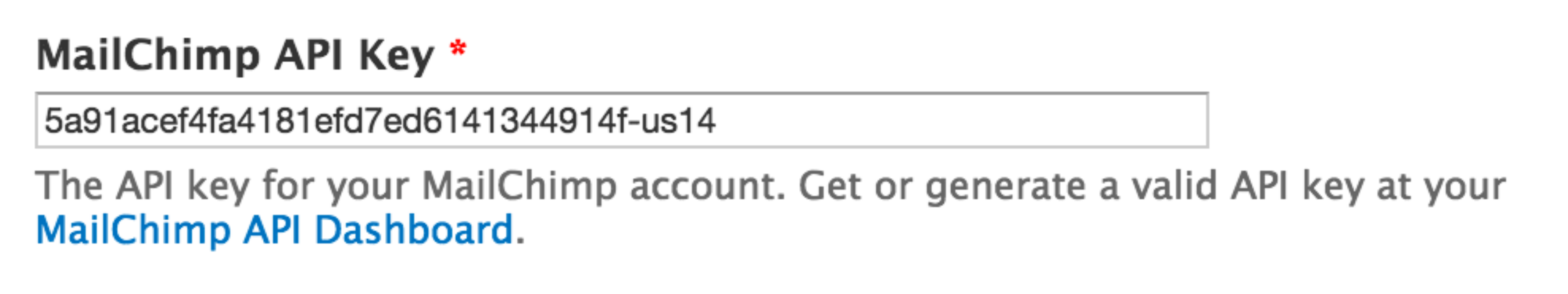
□ Base64-encoded

Check this if the key has been Base64-encoded. If the key should be used as-is, without Base64-decoding, leave this unchecked.

File location *

The location of the file in which the key will be stored. The path may be absolute (e.g., */etc/keys/foobar.key*), relative to the Drupal directory (e.g., *../keys/foobar.key*), or defined using a stream wrapper (e.g., *private://keys/foobar.key*).

MailChimp (before)

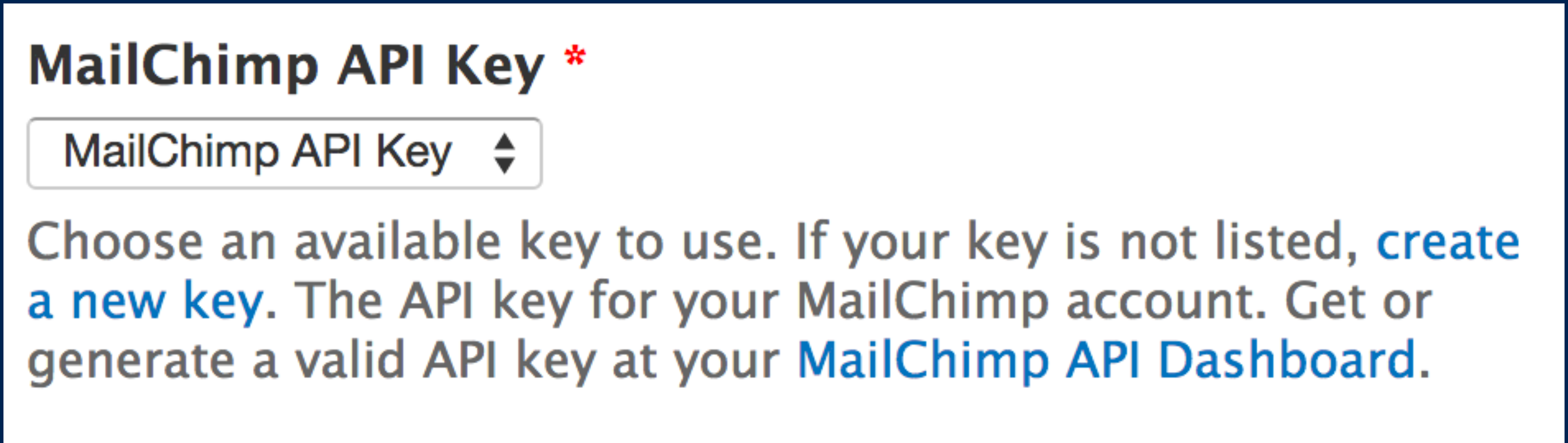


MailChimp API Key *

5a91acef4fa4181efd7ed6141344914f-us14

The API key for your MailChimp account. Get or generate a valid API key at your MailChimp API Dashboard.

MailChimp (after)



MailChimp API Key *

MailChimp API Key ⬍

Choose an available key to use. If your key is not listed, create a new key. The API key for your MailChimp account. Get or generate a valid API key at your MailChimp API Dashboard.

Encryption

- Encrypt module provides an API for encrypting and decrypting data
- Encryption methods are plugins
- Modules integrate with Encrypt to perform encryption on certain data within Drupal

Modules that integrate with Encrypt

- Encrypt Form API
- Field Encrypt
- Encrypted Files
- Townsend Security Key Connection
- Real AES
- Encrypt User

Encrypt for Drupal 8

- Will use Key for key management
- Development in GitHub: https://github.com/d8-contrib-modules/key

Other security-related modules

- Security Review
- Site Audit
- Password Policy
- Password Strength

- Secure Login
- CloudFlare
- Two-Factor Authentication
- Yubikey

Resources

- Full List Here: http://tiny.cc/d8-security-resources
- 10 Ways Drupal 8 Will Be More Secure (Acquia blog post)
- Drupal Security Group on groups.drupal.org
- DrupalCon sessions:
  - Drupal and Security: What You Need to Know
  - Securing Your Drupal Site: Advice for Site Builders and Coders

Drupal Security
Best Practices

By Mike Gifford,
OpenConcept Consulting Inc.

- Empowered
- Informed
- Comforted
- Confident

Thank you