

# Secure your app with 2FA

Rob Allen ~ @akrabat ~ November 2015

Your users' passwords will  
be leaked

(It might not even be your fault)

# Passwords have leaked from

Sony

Zappos

JP Morgan

Gap

Facebook

Twitter

eBay

AT&T

Adobe

Target

Blizzard

TalkTalk

Evernote

Yahoo!

Steam

Ubisoft

Kickstarter

Home Depot

TK Maxx

Vodafone

HP

AOL

Citigroup

WorldPay

Gmail

last.fm

Apple

SnapChat

Ubuntu

D&B

Formspring

Betfair

It will take 14 minutes\* to crack  
one of your users' passwords

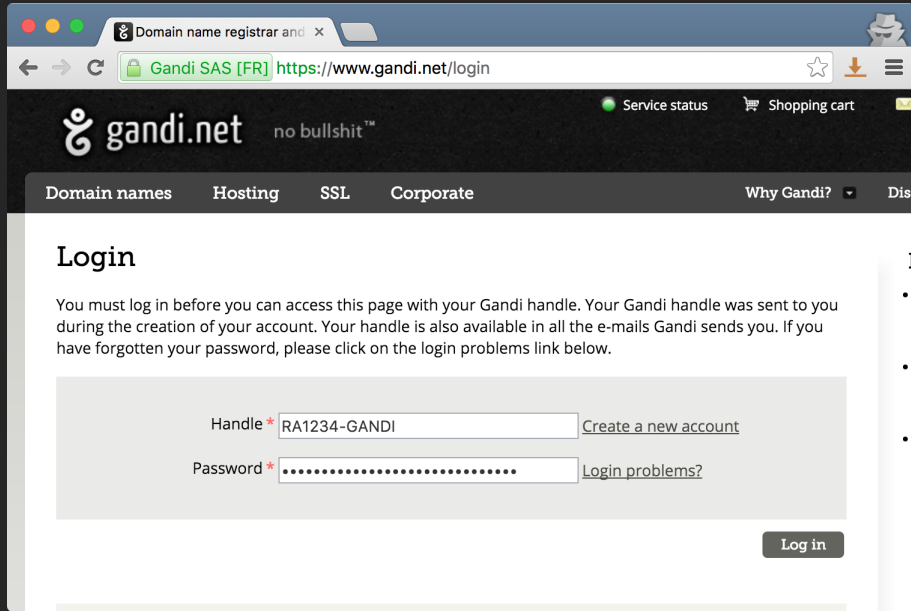
(English word, stored using bcrypt)

Two-factor authentication  
protects your users

# What is 2FA?



# Implementation on a website



The image shows a web browser window displaying the login page of Gandi.net. The browser's address bar shows the URL <https://www.gandi.net/login>. The page header includes the Gandi.net logo with the tagline "no bullshit™", a "Service status" indicator, and a "Shopping cart" icon. A navigation menu contains links for "Domain names", "Hosting", "SSL", "Corporate", "Why Gandi?", and "Disc".

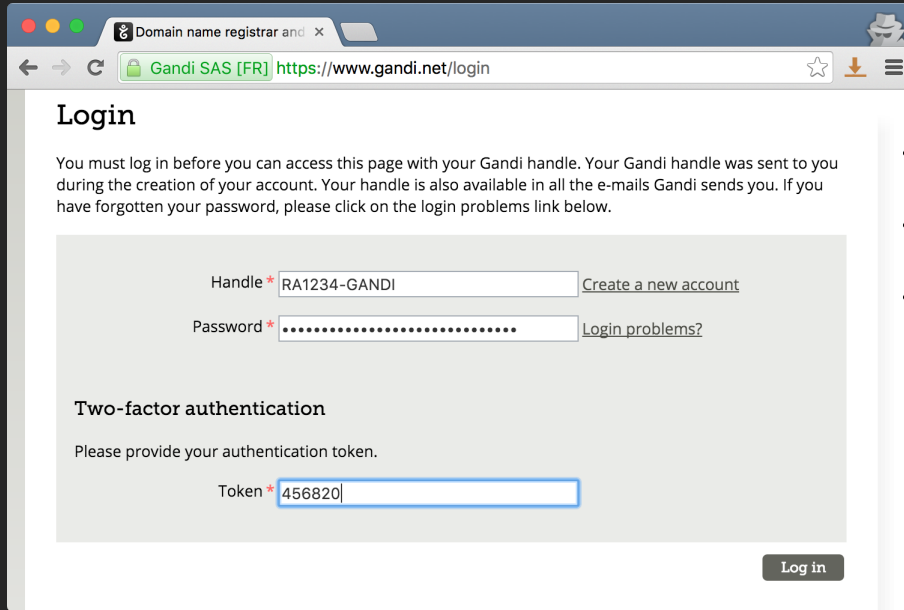
## Login

You must log in before you can access this page with your Gandi handle. Your Gandi handle was sent to you during the creation of your account. Your handle is also available in all the e-mails Gandi sends you. If you have forgotten your password, please click on the login problems link below.

Handle \*  [Create a new account](#)

Password \*  [Login problems?](#)

# Implementation on a website



The image shows a web browser window with the URL <https://www.gandi.net/login>. The page title is "Login". The main content area contains a login form with the following elements:

- Handle \***  [Create a new account](#)
- Password \***  [Login problems?](#)

Below the password field, there is a section for **Two-factor authentication** with the instruction "Please provide your authentication token." and a **Token \***

A **Log in** button is located at the bottom right of the form area.



How do we send the code?

# Email

- Used by Steam
- Wide adoption (everyone has an email address!)
- Likely failures: delivery problems, blocking, spam etc
- Usually slow!
- Same system as recover password...

# SMS

- Used by Twitter & LinkedIn
- Wide adoption
- But, SMS can be delayed & could cost to receive



# Physical device

- Used by banks, YubiKey, Blizzard, etc
- Small, long battery life
- But, expensive



# App

- Easy to use
- No Internet or cellular connection required
- App is free and trusted

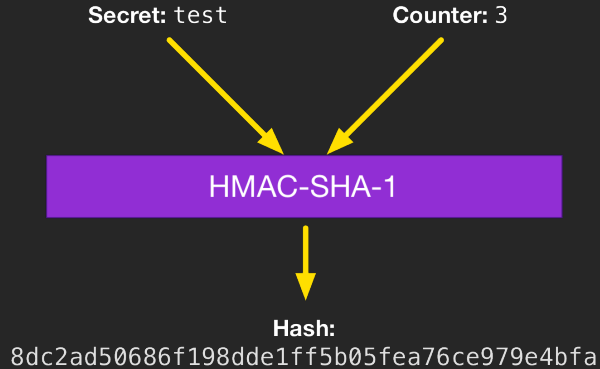


# One Time Password algorithms

# HOTP

- HMAC-based One-Time Password algorithm
- Computed from shared secret and counter
- New code each time you press the button
- RFC 4226

# HOTP algorithm: step 1





# HOTP algorithm: step 2

**Hash:**

8dc2ad50686f198dde1ff5b05fea76ce979e4bfa



Find lower 4 bits of last byte

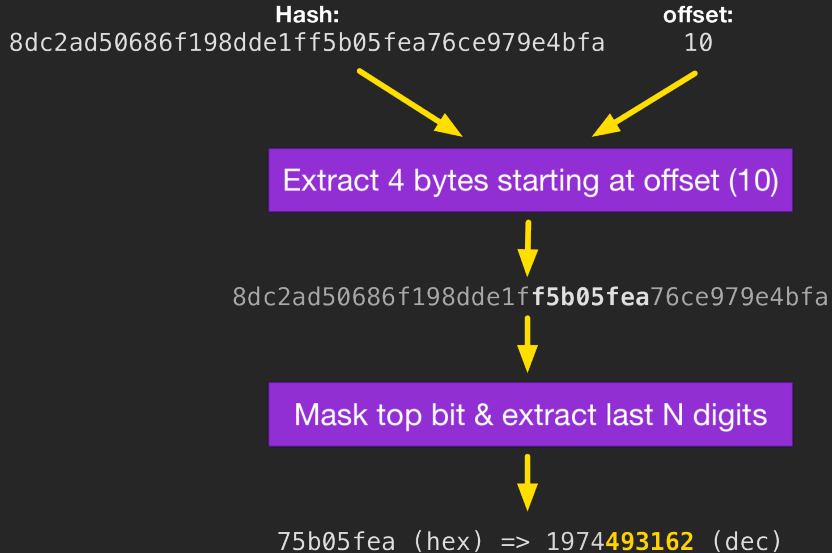


8dc2ad50686f198dde1ff5b05fea76ce979e4bfa

**offset:**

0x0A (hex) => 10 (decimal)

# HOTP algorithm: step 3



# HOTP in PHP

```
1 function hotp($secret, $counter)
2 {
3     $bin_counter = pack('J*', $counter);
4     $hash = hash_hmac('sha1', $bin_counter, $secret, true);
5
6     $offset = ord($hash[19]) & 0xf;
7
8     $bin_code =
9         ((ord($hash[$offset+0]) & 0x7f) << 24 ) |
10        ((ord($hash[$offset+1]) & 0xff) << 16 ) |
11        ((ord($hash[$offset+2]) & 0xff) << 8 ) |
12        (ord($hash[$offset+3]) & 0xff);
13
14     return $bin_code % pow(10, 6);
15 }
```

# Validation process

If the user's code matches, then increment counter by 1

If the user's code does not match, then look-ahead a little

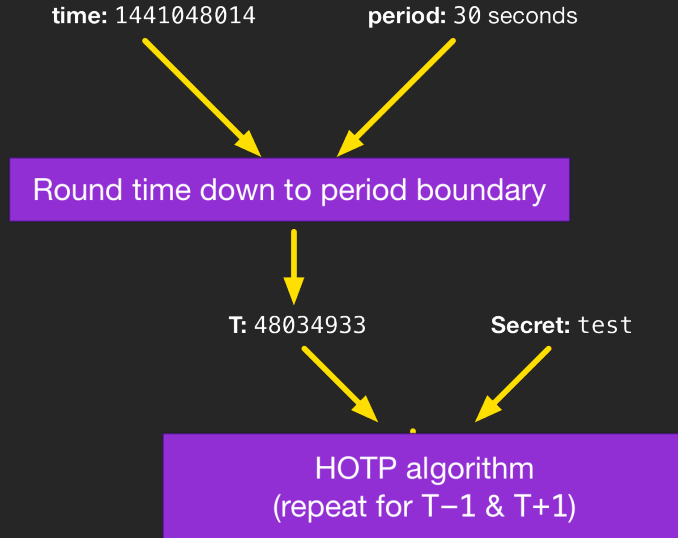
*Resync if can't find in look-ahead:*

1. Ask the user for two consecutive codes
2. Look ahead further from last known counter until the 2 codes are found
3. Limit look-ahead to minimise attack area. e.g. 400

# TOTP

- Time-based One-Time Password algorithm
- Computed from shared secret and current time
- Increases in 30 second intervals
- RFC 6238

# TOTP Algorithm



# TOTP in PHP

```
1 function totp($secret)
2 {
3     $counter = floor(time() / 30);
4
5     return hotp($secret, $counter);
6 }
```

# Implementing 2FA in your application



# Use a library!

```
$composer require sonata-project/google-authenticator
```

## Usage:

```
$g = new \Google\Authenticator\GoogleAuthenticator();
```

```
// create new secret and QR code
```

```
$secret = $g->generateSecret();
```

```
$qrCode = $g->getURL('rob', 'akrabat.com', $secret);
```

```
// validation of code
```

```
$g->checkCode($secret, $_POST['code']);
```

# Things to do

1. Registration of Authenticator
2. Check 2FA TOTP code during login

# Set up 2FA

User enables 2FA on their account

# Set up 2FA

User enables 2FA on their account



Site generates secret key and then displays QR code & confirmation field

# Set up 2FA

User enables 2FA on their account



Site generates secret key and then displays QR code & confirmation field



User adds to Authenticator

# Set up 2FA

User enables 2FA on their account



Site generates secret key and then displays QR code & confirmation field

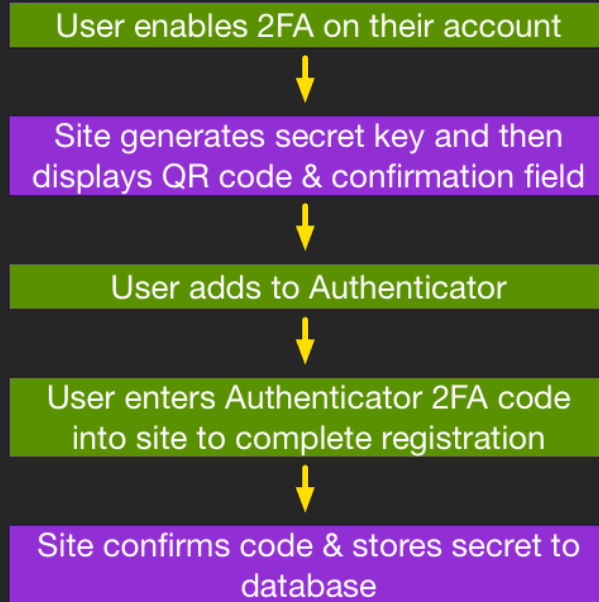


User adds to Authenticator



User enters Authenticator 2FA code into site to complete registration

# Set up 2FA



# Set up 2FA: code

```
1 $app->get('/setup2fa', function () use ($app) {
2     $user = $_SESSION['user'];
3
4     $g = new \Google\Authenticator\GoogleAuthenticator();
5     $secret = $g->generateSecret();
6     $qrCodeUrl = $g->getURL($user->getUsername(),
7         '2fa.dev', $secret);
8
9     $app->flash('secret', $secret);
10    $app->render('setup2fa.twig', [
11        'user'      => $_SESSION['user'],
12        'secret'    => $secret,
13        'qrCodeUrl' => $qrCodeUrl,
14    ]);
15 });
```



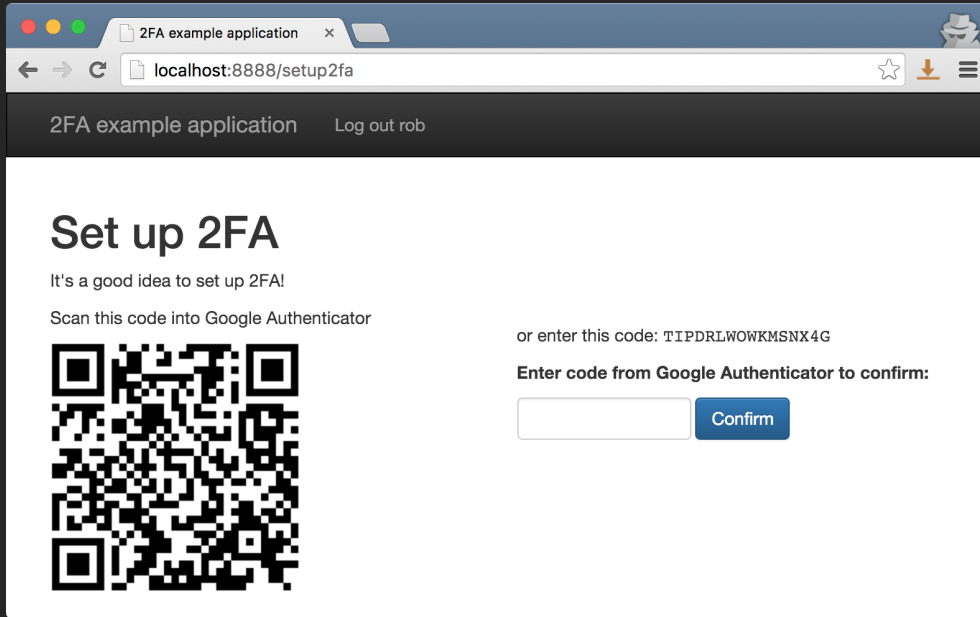
# Set up 2FA: template

```
1 <h1>Set up 2FA</h1>
2
3 <p>Scan this code into Google Authenticator:</p>
4 
5 <p>or enter this code: <tt>{{ secret }}</tt></p>
6
7 <p>Enter code from Google Authenticator to confirm:</p>
8 <form method="POST" action="/setup2fa">
9     <div class="form-group">
10         <input name="code" maxlength="6">
11         <button type="submit">Confirm</button>
12     </div>
13 </form>
```

# Set up 2FA: submission

```
1 $app->post('/setup2fa', function () use ($app) {
2     $secret = $app->environment['slim.flash']['secret'];
3     $code = $app->request->post('code');
4     $g = new \Google\Authenticator\GoogleAuthenticator();
5     if ($g->checkCode($secret, $code)) {
6         // code is valid - store into user record
7         $user = $_SESSION['user'];
8         $user->setSecret($secret);
9         $mapper = $app->userMapper;
10        $mapper->save($user);
11        $app->flash('message', 'Successfully set up 2FA');
12        $app->redirect('/');
13    }
14    $app->flash('error', 'Failed to confirm code');
15    $app->redirect('/setup2fa');
16 });
```

# Set up of 2FA




2FA example application [Log out rob](#)

## Set up 2FA

It's a good idea to set up 2FA!

Scan this code into Google Authenticator



or enter this code: `TIPDRLWOWKMSNX4G`

Enter code from Google Authenticator to confirm:

# Logging in

After login form, prompt for 2FA code

# Logging in

After login form, prompt for 2FA code



User enters 2FA code

# Logging in

After login form, prompt for 2FA code

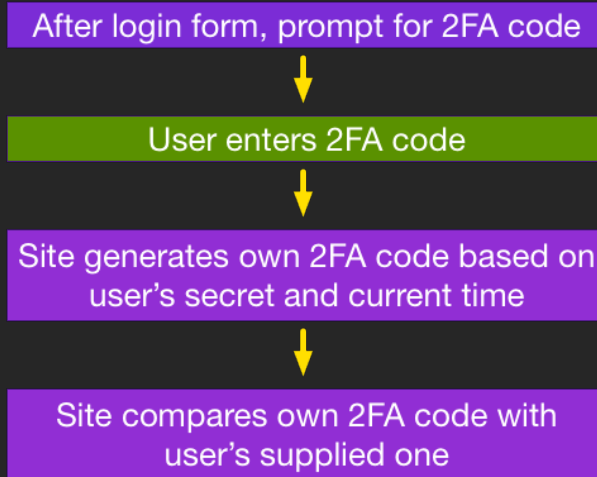


User enters 2FA code

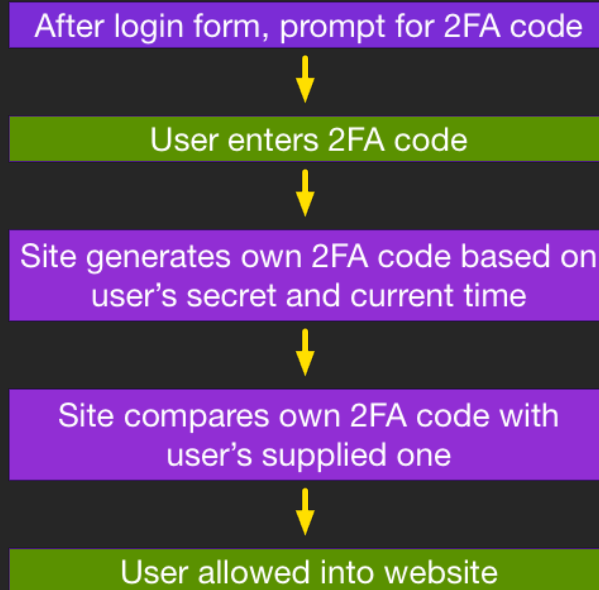


Site generates own 2FA code based on user's secret and current time

# Logging in



# Logging in





# Login: display username/pwd

```
1 $app->get('/login', function () use ($app) {  
2     $app->render('login.twig');  
3 });
```

# Login: display username/pwd

```
1 <h1>Login</h1>
2
3 <form method="POST" action="/login" class="form-horizontal">
4   <div class="form-group">
5     <label for="username">Username</label>
6     <div><input type="text" id="username" name="username"></div>
7   </div>
8   <div class="form-group">
9     <label for="password">Password</label>
10    <div><input type="password" id="p :)
11    assword" name="password"></div>
12  </div>
13  <div class="form-group">
14    <button type="submit">Log in</button>
15  </div>
16 </form>
```

# Login: process username/pwd

```
1 $app->post('/login', function () use ($app, $mapper) {
2     $username = $app->request->post('username');
3     $password = $app->request->post('password');
4     $user = $mapper->load($username);
5     $valid = password_verify($password, $user->getPassword());
6     if ($valid) {
7         if ($user->getSecret()) {
8             $_SESSION['user_in_progress'] = $user;
9             $app->redirect('/auth2fa');
10        }
11        $_SESSION['user'] = $user;
12        $app->redirect('/');
13    }
14    $app->flash('error', 'Failed to log in');
15    $app->redirect('/login');
16 });
```

# Login: 2FA code form

```
1 $app->get('/auth2fa', function () use ($app) {  
2     $app->render('auth2fa.twig');  
3 });
```

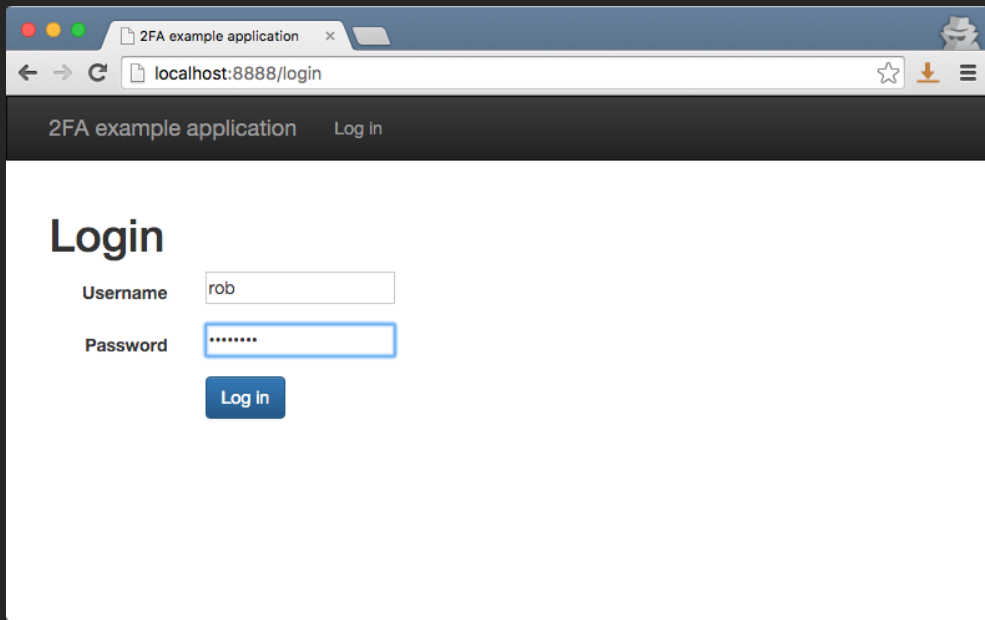
auth2fa.twig:

```
1 <h1>2FA authentication</h1>  
2  
3 <p>Enter the code from Google Authenticator  
4   to complete login:</p>  
5 <form method="POST" action="/auth2fa">  
6     <div class="form-group">  
7         <input name="code" maxlength="6">  
8         <button type="submit">Submit</button>  
9     </div>  
10 </form>
```

# Login: process 2FA code

```
1 $app->post('/auth2fa', function () use ($app) {
2     $user    = $_SESSION['user_in_progress'];
3     $secret  = $user->getSecret();
4     $code    = $app->request->post('code');
5
6     $g = new \Google\Authenticator\GoogleAuthenticator();
7     if ($g->checkCode($secret, $code)) {
8         // code is valid!
9         $_SESSION['user'] = $_SESSION['user_in_progress'];
10        unset($_SESSION['user_in_progress']);
11        $app->flash('message', 'Successfully logged in');
12        $app->redirect('/');
13    }
14
15    $app->flash('error', 'Failed to confirm code');
16    $app->redirect('/auth2fa');
17 });
```

# Login



The image shows a web browser window with the following elements:

- Browser tab: 2FA example application
- Address bar: localhost:8888/login
- Page title: 2FA example application
- Page content:
  - Header: 2FA example application Log in
  - Section title: Login
  - Form fields:
    - Username: rob
    - Password: masked with dots
  - Submit button: Log in

# 2FA code

2FA example application    Log in

## 2FA authentication

Enter the code from Google Authenticator to complete login:

# Round out solution

- Prevent brute force attacks
- Consider adding a “remember this browser” feature
- Need a solution for a lost/new phone

Example project: <https://github.com/akrobat/slim-2fa>



# Hardware OTP: YubiKey



# Operation

1. Insert YubiKey into USB slot
2. Select input field on form
3. Press button to fill in OTP field
4. Server validates OTP with YubiCloud service

# Yubikey's OTP

cdccccetfjnfjgkkgudlkcbjnfnfrkhkbelbhfvnhcj  
cdccccetfjnfuuhtrebhjekcciljdbifgfrlccbnjhkf  
cdccccetfjnf`ljrhetskvi`cgddkenbtuiknktejgngvb



Public id



Yubico OTP

6 byte private identity field  
Counter  
Timer field  
Random number  
CRC16 checksum

# Coding it

```
$composer require enygma/yubikey
```

## Usage:

```
$v = new \Yubikey\Validate($apiKey, $clientId);  
$response = $v->check($_POST['yubikey_code']);  
  
if ($response->success() === true) {  
    // allow into website  
}
```

# Login: process yubikey code

```
1 $app->post('/auth2fa', function () use ($app) {
2     $apiKey = $app->settings['yubikey']['api_key'];
3     $clientId = $app->settings['yubikey']['client_id'];
4
5     $v = new \Yubikey\Validate($apiKey, $clientId);
6     $response = $v->check($app->request->post('code'));
7
8     if ($response->success() === true) {
9         // Successfully logged in
10        $_SESSION['user'] = $_SESSION['user_in_progress'];
11        unset($_SESSION['user_in_progress']);
12        $app->redirect('/');
13    }
14    $app->flash('error', 'Failed to confirm code');
15    $app->redirect('/auth2fa');
16 });
```

That's all there is to it!

Pre-built plugins

# Pre-built plugins

Drupal:

- Two-factor Authentication
- Yubikey

WordPress:

- Two-Factor
- (many others!)

Joomla:

- Built-in!



# Drupal

The screenshot shows a web browser window displaying the Drupal.org website. The address bar shows the URL <https://www.drupal.org/project/tfa>. The page header is blue with the Drupal logo and navigation links: Get Started, Community, Documentation, Support, Download & Extend, Jobs, Marketplace, and About. A search bar is located in the top right corner. Below the header, there are buttons for "Drupal Homepage" and "Log in / Register". The main content area is titled "Download & Extend" and has sub-tabs for "Download & Extend Home", "Drupal Core", "Distributions", "Modules", and "Themes". The "Modules" tab is selected, showing the "Two-factor Authentication (TFA)" module page. The page includes a "View" button, "Version control" and "Automated Testing" links, and a post by "coltrane" from March 21, 2011. The main text describes the TFA module as a base module for providing two-factor authentication. A sidebar on the right contains sections for "Maintainers for Two-factor Authentication (TFA)" and "Issues for Two-factor Authentication (TFA)".

Two-factor Authentication (TFA)

[View](#) [Version control](#) [Automated Testing](#)

Posted by [coltrane](#) on *March 21, 2011 at 3:19am*

Second-factor authentication for Drupal sites. Drupal provides authentication via something you *know* -- a username and password while TFA module adds a second step of authentication with a check for something you *have* -- such as a code sent to (or generated by) your mobile phone.

TFA is a base module for providing two-factor authentication for your Drupal site. As a base module, TFA handles the work of integrating with Drupal, providing flexible and well tested interfaces to enable your choice of various two-factor authentication solutions like Time-based One Time Passwords (TOTP), SMS-delivered codes, pre-generated codes, or integrations with third-party services like Authy, Duo and others.

Read the [TFA module documentation](#) or read more about the [theory of two-factor authentication](#) in my [Drupal Watchdog](#) article.

**Features**

- Pluggable – Supports multiple methods of 2nd factor verification and can work with any number of 3rd party systems

**Maintainers for Two-factor Authentication (TFA)**

[coltrane](#) – 70 commits  
last: 1 month ago, first: 4 years ago

[greggles](#) – 2 commits  
last: 1 year ago, first: 1 year ago

[View all committers](#)  
[View commits](#)

**Issues for Two-factor Authentication (TFA)**

To avoid duplicates, please search before submitting a new issue.  
[Advanced search](#)

# Set up

The screenshot shows a web browser window displaying the configuration page for Two-factor Authentication in Drupal. The browser's address bar shows the URL `drupal.akrabat.com/overlay-admin/config/people/tfa`. The page header includes navigation links for Dashboard, Content, Structure, Appearance, People, Modules, Configuration, Reports, and Help. The user is logged in as 'Hello roballen' and can click 'Log out'. Below the header, there are links for 'Add content', 'Find content', and 'Edit shortcuts'. The main content area is titled 'Two-factor Authentication' and includes a breadcrumb trail: 'Home > Administration > Configuration > People'. A list of 'AVAILABLE PLUGINS' is shown, including TOTP (validation) - active validator, Trusted Browsers (login) - active login, Recovery Codes (validation) - unused, Twilio SMS (validation, send) - unused, and Help page (validation) - unused. The 'Enable TFA' checkbox is checked, and the text below it reads 'Enable TFA for account authentication.' The 'Default validation plugin' dropdown menu is set to 'TOTP', with a note that this plugin will be used as the default TFA process. At the bottom, there is a section for 'VALIDATION FALLBACK PLUGINS' with a note that fallback plugins will not be active if not set up, and a checkbox for 'Recovery Codes' is visible.

Two-factor Authentication

Home > Administration > Configuration > People

**AVAILABLE PLUGINS**

- TOTP (*validation*) – active validator
- Trusted Browsers (*login*) – active login
- Recovery Codes (*validation*) – unused
- Twilio SMS (*validation, send*) – unused
- Help page (*validation*) – unused

Enable TFA  
Enable TFA for account authentication.

**Default validation plugin**

TOTP

Plugin that will be used as the default TFA process.

**VALIDATION FALLBACK PLUGINS**

Fallback plugins and order. Note, if a fallback plugin is not setup for an account it will not be active in the TFA form.

Recovery Codes

# Set up

TFA setup - Application | Ro... x +

drupal.akrabat.com/User/1/security/tfa/app-setup

Search

Dashboard Content Structure Appearance People Modules Configuration Reports Help

Hello roballen Log out

Add content Find content Edit shortcuts

## TFA setup - Application


Install authentication code application on your mobile or desktop device:

- [Google Authenticator \(Android/iPhone/BlackBerry\)](#)
- [Authy \(Android/iPhone\)](#)
- [Authenticator \(Windows Phone\)](#)
- [FreeOTP \(Android\)](#)
- [GAuth Authenticator \(desktop\)](#)

The two-factor authentication application will be used during this setup and for generating codes during regular authentication. If the application supports it, scan the QR code below to get the setup code otherwise you can manually enter the text code.

78Z4DCHUQ4DLOZ06

Enter this code into your two-factor authentication app or scan the QR code below.

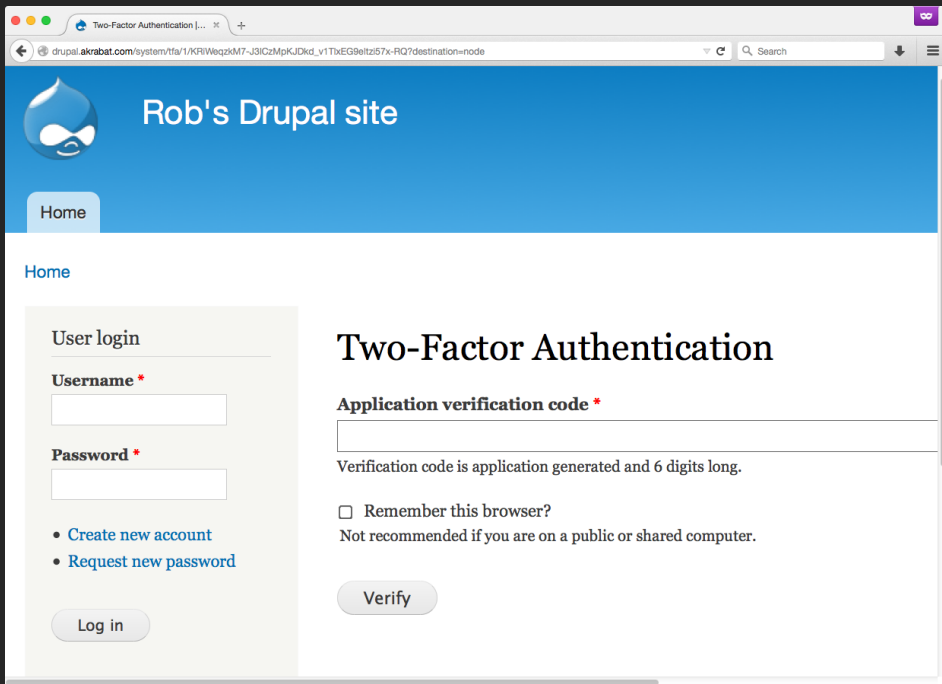


Application verification code \*

A verification code will be generated after you scan the above QR code or manually enter the setup code. The verification code is six digits long.

Verify and save Cancel

# Log in



The screenshot shows a web browser window with the address bar containing a URL for a Two-Factor Authentication page on a Drupal site. The page header features the Drupal logo and the text 'Rob's Drupal site'. Below the header is a navigation bar with a 'Home' button. The main content area is titled 'Home' and contains two primary sections: a 'User login' form and a 'Two-Factor Authentication' section.

**User login**

**Username \***

**Password \***

- [Create new account](#)
- [Request new password](#)

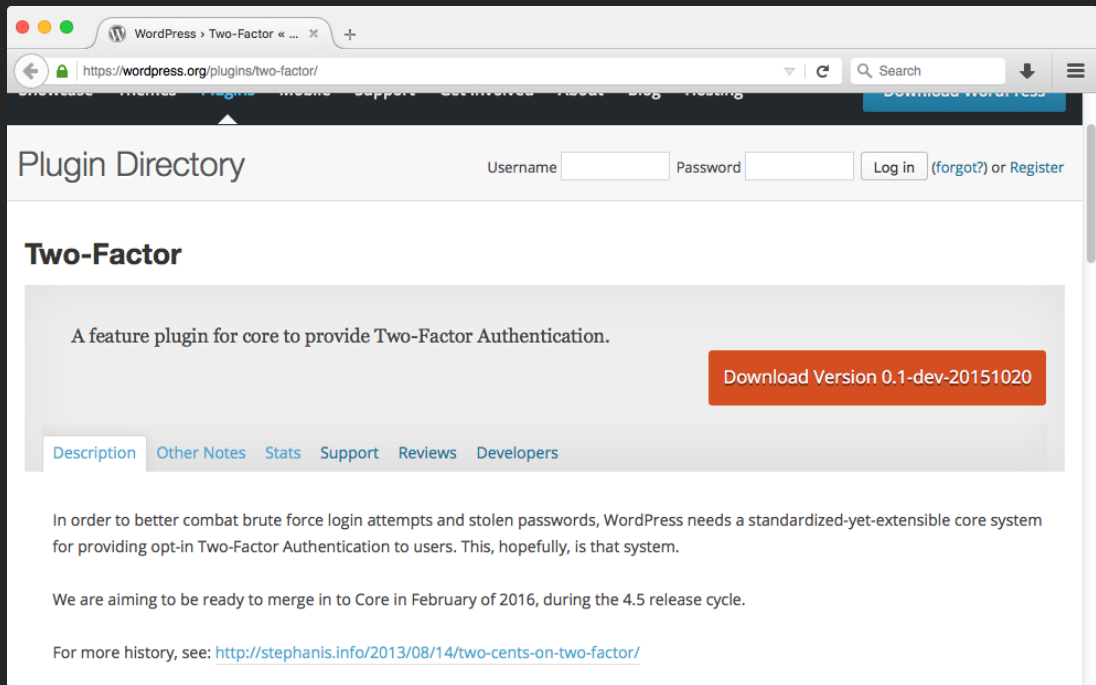
**Two-Factor Authentication**

**Application verification code \***

Verification code is application generated and 6 digits long.

Remember this browser?  
Not recommended if you are on a public or shared computer.

# WordPress



The image shows a browser window displaying the WordPress Plugin Directory page for the 'Two-Factor' plugin. The browser's address bar shows the URL 'https://wordpress.org/plugins/two-factor/'. The page header includes the 'Plugin Directory' title and a login section with 'Username' and 'Password' input fields, a 'Log in' button, and links for '(forgot?) or Register'. The main content area features the plugin title 'Two-Factor' and a description: 'A feature plugin for core to provide Two-Factor Authentication.' A prominent orange button labeled 'Download Version 0.1-dev-20151020' is positioned to the right of the description. Below the description is a navigation bar with tabs for 'Description', 'Other Notes', 'Stats', 'Support', 'Reviews', and 'Developers'. The 'Description' tab is active, showing a paragraph about the plugin's purpose: 'In order to better combat brute force login attempts and stolen passwords, WordPress needs a standardized-yet-extensible core system for providing opt-in Two-Factor Authentication to users. This, hopefully, is that system.' A second paragraph states: 'We are aiming to be ready to merge in to Core in February of 2016, during the 4.5 release cycle.' At the bottom, there is a link for more history: 'For more history, see: <http://stephanis.info/2013/08/14/two-cents-on-two-factor/>'.

WordPress > Two-Factor

https://wordpress.org/plugins/two-factor/

Plugin Directory

Username  Password  Log in (forgot?) or Register

## Two-Factor

A feature plugin for core to provide Two-Factor Authentication.

Download Version 0.1-dev-20151020

Description Other Notes Stats Support Reviews Developers

In order to better combat brute force login attempts and stolen passwords, WordPress needs a standardized-yet-extensible core system for providing opt-in Two-Factor Authentication to users. This, hopefully, is that system.


We are aiming to be ready to merge in to Core in February of 2016, during the 4.5 release cycle.

For more history, see: <http://stephanis.info/2013/08/14/two-cents-on-two-factor/>

# Set up

- Dashboard
- Posts
- Media
- Pages
- Comments
- Appearance
- Plugins 1
- Users**
- All Users
- Add New
- Your Profile
- Tools
- Settings
- Collapse menu

## Two-Factor Options

Enabled	Primary	Name
<input type="checkbox"/>	<input type="radio"/>	Email Authentication codes will be sent to rob@akrabat.com.
<input checked="" type="checkbox"/>	<input checked="" type="radio"/>	<a href="#">Time Based One-Time Password (Google Authenticator)</a> <a href="#">View Options →</a>  HZLGSOBEKRPTY3LFMJPCU4RILJIUAOSS Please scan the QR code or manually enter the key, then enter an authentication code from your app in order to complete setup Authentication Code: <input type="text"/>
<input type="checkbox"/>	<input type="radio"/>	FIDO U2F You need to register security keys such as Yubikey.
<input checked="" type="checkbox"/>	<input type="radio"/>	<a href="#">Backup Verification Codes (Single Use)</a> <input type="button" value="Generate Verification Codes"/> 10 unused codes remaining.

# Log in



Username

Password

Remember Me

Log In

[Lost your password?](#)

[← Back to Rob Allen's DevNotes](#)

# Log in



Authentication Code:

Authenticate

Or, use your backup method: Backup  
Verification Codes (Single Use) →

← [Back to Rob Allen's DevNotes](#)



# Joomla

Rob's Joomla! - Administrati... x +

localhost/joomla/administrator/index.php?option=com\_users&view=user&layout=edit&id=567

System Users Menus Content Components Extensions Help

Rob's Joomla! Joomla!

User Manager: Edit Profile

Save Save & Close Save & New Close Help

### Super User

Account Details Assigned User Groups Basic Settings Two Factor Authentication

Authentication Method YubiKey

This feature allows you to use a YubiKey secure hardware token for two factor authentication. In addition to your username and password you will also need to insert your YubiKey into your computer's USB port, click inside the Secret Key area of the site's login area and touch YubiKey's gold disk. The secret code generated by your YubiKey is unique to your device and changes constantly. This provides extra protection against hackers logging in to your account even if they were able to get hold of your password.

### Set up

Please insert your YubiKey device into your computer's USB port. Click on the Security Code field below. Then touch the gold disk on your YubiKey device for one second. Afterwards, please save your user profile. If the code generated by your YubiKey is validated by YubiCloud the Two Factor Authentication feature will be enabled and this YubiKey will be linked with your user account.

Security Code

### One time emergency passwords

If you do not have access to your two factor authentication device you can use any of the following passwords instead of a regular security code. Each one of

View Site 1 Visitor 1 Admin 0 Log out Joomla! 3.4.3 — © 2015 Rob's Joomla!

# Log in



The image shows the Joomla! login interface. At the top left is the Joomla! logo, which consists of four interlocking rings in green, orange, red, and blue, followed by the text "Joomla!" with a registered trademark symbol. Below the logo are three input fields, each with a small icon on the left and a help icon on the right. The first field is labeled "Username" with a person icon. The second field is labeled "Password" with a lock icon. The third field is labeled "Secret Key" with a star icon. Below these fields is a large blue button with a white lock icon and the text "Log in". At the bottom center of the dark blue background is a small white Joomla! logo.

 Joomla!®

?

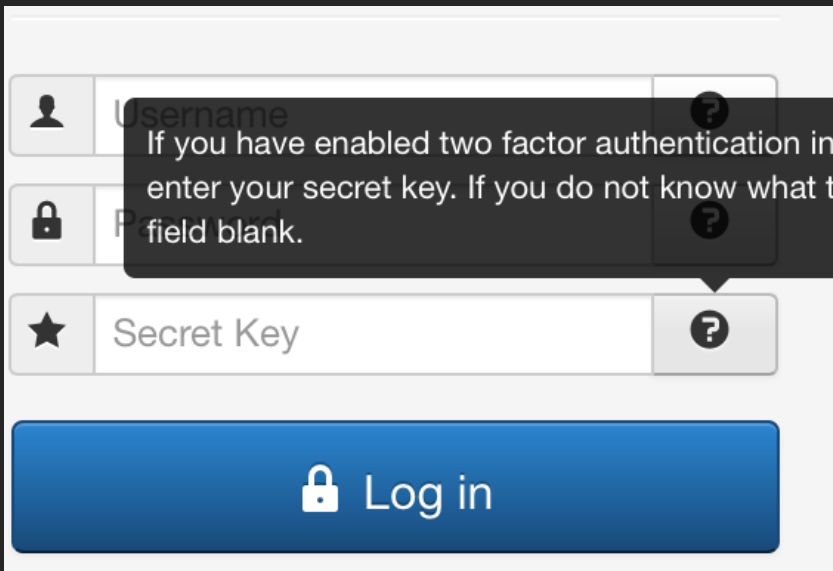
?

?




 Log in




# That secret key field



The image shows a login form with three input fields and a 'Log in' button. The fields are: 'Username' (with a person icon), 'Password' (with a lock icon), and 'Secret Key' (with a star icon). Each field has a question mark icon in its right-hand corner. A dark grey tooltip box is overlaid on the 'Username' and 'Password' fields, containing the text: 'If you have enabled two factor authentication in your user account please enter your secret key. If you do not know what this means, you can leave this field blank.' Below the fields is a blue button with a white lock icon and the text 'Log in'.

	Username	?
	Password	?
	Secret Key	?

 Log in

If you have enabled two factor authentication in your user account please enter your secret key. If you do not know what this means, you can leave this field blank.

To sum up

To sum up

Two-factor authentication isn't hard!

# Questions?

<https://joind.in/14761>

Rob Allen - <http://akrabat.com> - @akrabat

Thank you!

<https://joind.in/14761>

Rob Allen - <http://akrabat.com> - @akrabat