

Ofuscar Palabras Con ROT13 En WordPress

Existen una gran cantidad de **algoritmos para el cifrado de datos** que pueden ser usados por los programadores. Uno de los menos conocidos y comunes es el **cifrado ROT13**, un derivado de la técnica de [cifrado César](#). En este tutorial, aprenderás sobre el cifrado ROT13 y cómo funciona. Además también aprenderás a **desarrollar un plugin de WordPress** que escanea una entrada para determinadas palabras en una «lista negra» y reemplazarlas con cualquier otra palabra cifrada con el método ROT13.

Si eres el propietario de un blog en el que existen varios autores o cierto grupo de personas tienen el privilegio de publicar artículos, un plugin que encripta o elimina palabras inapropiadas podría ser gran utilidad ;)

¿Qué es ROT13?

ROT13 (abreviatura de «rotate by 13 places») es una **técnica de cifrado simple**, sustituye cada letra con la siguiente en el lugar 13 hacia adelante o hacia atrás a lo largo del alfabeto. Por lo tanto, A se convierte en N, B en O, y de esta manera podrás tener un contenido cifrado con esta técnica.

Una gran ventaja de ROT13, es que es «auto-inversa», es decir que el mismo algoritmo se aplica para **cifrar y descifrar datos**. A continuación puedes ver una tabla ROT13 para una fácil referencia.



Si cifró el dominio **HostDimeColombia** en ROT13, el resultado sería UbfgQvzrPbybzovn, y la frase «¿Por qué el pollo cruzó el

camino?» Se convertiría en «¿Cbe dhr ry cbyyb pehmb ry pnzvab?» Ten en cuenta que **sólo las letras del alfabeto** se ven afectados por ROT13. Números, símbolos, espacios en blanco y todos los demás caracteres no son modificados.

Transformando cadenas a ROT13 en PHP

PHP incluye la **función** `str_rot13()`, para convertir una cadena usando el **método de cifrado ROT13**. Para codificar texto en ROT13, pase el texto como argumento de la función.

```
[php]
```

```
<?php
```

```
echo str_rot13('HostDimeColombia'); //Tendrá como salida  
-> UbfgQvzrPbybzovn
```

```
echo str_rot13('Conoce nuestros planes en servicios web'); //  
Obtendríamos -> Pbabpr ahrfgebfb cynarf ra freivpvbf jro
```

```
[/php]
```

Usando ROT13 En WordPress

Como ya hemos visto anteriormente, el [desarrollo de un plugin para WordPress](#) es bastante fácil. Solo se debe saber lo que se desea hacer, seguir algunos pasos y listo. Tranquilo, te mostraremos como puedes desarrollar tu propio plugin, luego

podrás modificarlo a tu gusto y necesidades.

Configuración Del Plugin

En primer lugar, incluir un encabezado de archivo del plugin.

```
[php]

<?php

/*
Plugin Name: Rot13 Words Blacklist
Plugin URI: URL que desees configurar
Description: Descripcion del plugin
Version: 1.0
Author: Nombre del autor
Author URI: URL perfil autor
Text Domain: rot13
Domain Path: /lang/
License: GPL2
*/

[/php]
```

Como se ha mencionado, el plugin tendrá una sección para una «lista de palabras negra», para esto usaremos un campo textarea que obtenga y almacene las palabras de la lista negra en la base de datos de WordPress (específicamente la tabla de opciones). A continuación se muestra una captura de pantalla de lo se vería en la opción de lista negra:



Ahora que sabemos lo que la página de opciones se verá así, vamos a construir utilizando WordPress 'Configuración API

Desarrollando La Página De Ajustes

En primer lugar, creamos un elemento de submenú en el menú principal «Ajustes», para esto usamos **add_options_page()**, con su función principal añadirás la acción al menú principal.

[php]

```
add_action( 'admin_menu', 'rot13_plugin_menu' );

/**
 * Add submenu to main Settings menu
 */
function rot13_plugin_menu() {
    add_options_page(
        __( 'Rot13 Blacklisted Words', 'rot13' ),
        __( 'Rot13 Blacklisted Words', 'rot13' ),
        'manage_options',
        'rot13-words-blacklist',
        'rot13_plugin_settings_page'
    );
}
```

[/php]

El quinto parámetro de `add_options_page()`, es el nombre de la función (`rot13_plugin_settings_page`), que llama la salida de los contenidos de la página. A continuación se muestra el código para `rot13_plugin_settings_page()`.

[php]

```
/**
 * Output the contents of the settings page.
 */
function rot13_plugin_settings_page() {
```

```

echo '<div class="wrap">';
echo '<h2>', __( 'Rot13 Blacklisted Words', 'rot13' ),
'</h2>';
echo '<form action="options.php" method="post">';
do_settings_sections( 'rot13-words-blacklist' );
settings_fields( 'rot13_settings_group' );
submit_button();
}

[/php]

```

A continuación, añadimos una nueva sección a la página «Configuración» con **add_settings_section()**. El campo textarea se añade a esta sección con **add_settings_field()**. Por último, los ajustes están registrados en **register_setting()**. A continuación se muestra el código para **add_settings_section()**, **add_settings_field()** y **register_setting()**.

```

[php]

// Add the section
add_settings_section(
    'rot13_setting_section',
    »,
    'rot13_setting_section_callback_function',
    'rot13-words-blacklist'
);
// Add the textarea field to the section.
add_settings_field(
    'blacklisted_words',
    __( 'Blacklisted words', 'rot13' ),
    'rot13_setting_callback_function',
    'rot13-words-blacklist',
    'rot13_setting_section'
);

// Register our setting so that $_POST handling is done for us
register_setting(
    'rot13_settings_group',

```

```
'rot13_plugin_option', 'sanitize_text_field' );
```

```
[/php]
```

Las tres funciones, deberán conservarse cerrados en una función y luego añadirse a la acción `admin_init`, justo de la siguiente manera:

```
[php]
```

```
/**
 * Hook the Settings API to 'admin_init' action
 */
function rot13_settings_api_init() {
    // Add the section
    add_settings_section(
        'rot13_setting_section',
        »,
        'rot13_setting_section_callback_function',
        'rot13-words-blacklist'
    );
    // Add the textarea field to the section
    add_settings_field(
        'blacklisted_words',
        __( 'Blacklisted words', 'rot13' ),
        'rot13_setting_callback_function',
        'rot13-words-blacklist',
        'rot13_setting_section'
    );

    // Register our setting so that $_POST handling is done for us
    register_setting( 'rot13_settings_group',
        'rot13_plugin_option', 'sanitize_text_field' );
}
```

```
add_action( 'admin_init', 'rot13_settings_api_init' );
```

```
[/php]
```

A continuación está el código para la función `rot13_setting_callback_function()` y `rot13_setting_section_callback_function()`, esto es para la salida del campo textarea y la descripción del campo, respectivamente.

```
[php]
```

```
/**
 * Add a description of the field to the top of the section
 */
function rot13_setting_section_callback_function() {
echo '<p>' . __( 'Enter a list of words to blacklist,
separated by commas (,)', 'rot13' ) . '</p>';
}

/**
 * Callback function to output the textarea form field
 */
function rot13_setting_callback_function() {
echo '<textarea rows="10" cols="60" name="rot13_plugin_option"
class="code">' . esc_textarea( get_option(
'rot13_plugin_option' ) ) . '</textarea>';
}
```

```
[/php]
```

En este punto, hemos finalizado el desarrollo de la página de configuración para el plugin. Lo siguiente es conseguir que el plugin detecte palabras de la lista negra y las cifre con ROT13.

Detectando Palabras De

La Lista Negra Y Cifrando Con ROT13

He aquí un resumen de será el proceso para **detectar palabras de la lista negra** en un post de WordPress:

- El contenido de un post se descompone en palabras individuales y se guardan en un array (`$post_words`).
- Las palabras de la lista negra que se almacenaron por el plugin para la base de datos se obtienen. Ellos, también, se descomponen en palabras individuales y se guardan en un array (`$blacklisted_words`).
- Iteramos sobre la matriz de `$post_words` y el registro para cualquier palabra que está en la lista negra.
- Si se encuentra una palabra en la lista negra, entonces el método `str_rot13()` codifica en ROT13.

Es el momento de crear la función de PHP (`rot13_filter_post_content()`) que filtra el contenido de un artículo y luego detecta palabras de la lista negra y las cifra en ROT13. A continuación se muestra el código para el filtro del mensaje.

[php]

```
/**
 * Encrypt every blacklisted word in ROT13
 *
 * @param $content string post content to filter
 *
 * @return string
 */
function rot13_filter_post_content( $content ) {

// Get the words marked as blacklisted by the plugin
$blacklisted_words = esc_textarea( get_option(
```



```

'rot13_plugin_option' ) );

// If no blacklisted word are defined, return the post's
content.
if ( empty( $blacklisted_words ) ) {
return $content;
}

else {

// Ensure we are dealing with "posts", not "pages" or any
other content type.
if ( is_singular( 'post' ) ) {

// Confine each word in a post to an array
$post_words = preg_split( "/\b/", $content );

// Break down the post's contents into individual words
$blacklisted_words = explode( ',', $blacklisted_words );

// Remove any leading or trailing white space
$blacklisted_words = array_map(
function ( $arg ) {
return trim( $arg );
},

$blacklisted_words
);
// Iterate over the array of words in the post
foreach ( $post_words as $key => $value ) {

// Iterate over the array of blacklisted words
foreach ( $blacklisted_words as $words ) {

// Compare the words, being case-insensitive
if ( strcasecmp( $post_words[ $key ], $words ) == 0 ) {

// Encrypt any blacklisted word
$post_words[ $key ] = '<del>' . str_rot13( $value ) .
'</del>';

```

```

}
}
}

// Convert the individual words in the post back into a string
or text
$content = implode( », $post_words );
}

return $content;
}
}
add_filter( 'the_content', 'rot13_filter_post_content' );

[/php]

```

La función `is_singular` («post») para el tag condicional, asegura que se trata de un artículo, y no de una página o cualquier otro tipo de contenido. Con `preg_split()`, separamos el contenido del artículo en palabras individuales y las almacenamos como una matriz mediante la búsqueda por una expresión regular. La lista de palabras de la lista negra se recupera de la base de datos utilizando el método `get_option()`, con `rot13_plugin_option` como el nombre del parámetro. Desde la captura de pantalla de la página de configuración del plugin de arriba y la descripción del campo textarea, podemos ver que las palabras de la lista negra están separados por comas, nuestro delimitador ;)

Un cierre se aplica a la matriz a través del arreglo `$blacklisted_words` por medio de `array_map()` que elimina espacios blancos iniciales y finales de los valores de matriz (las palabras en la lista negra individuales). La construcción de `foreach` itera sobre las palabras del artículo y comprueba si la palabra está en la matriz de palabras en la lista negra. Cualquier palabra en la lista negra que sea detectada se en ROT13 y es encerrada en una etiqueta `<del>`.

El array `$post_words` se convierte de nuevo en una cadena o

texto y posteriormente será devuelto. Finalmente, la función se añade a la acción del filtro `the_content`. A continuación se muestra una captura de pantalla de un puesto con las palabras en la lista negra como «love» y «forever».



Finalmente

Ofuscar palabras con ROT13, es algo bastante simple que pueda ser descifrado fácilmente. Por lo tanto, nunca se debe utilizar para el cifrado de datos sensibles. Incluso si no terminas usando el plugin, los conceptos que has aprendido en la creación se puede aplicar a diferentes situaciones, como la ofuscación para cifrar palabras inadecuadas (como palabras de obscenas) en ROT13. Si conoces otro método simple, a parte de los ya conocidos, compártelo en un comentario ;)