RIPSTECH

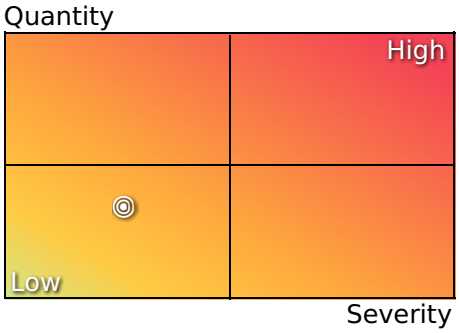
SECURITY ANALYSIS REPORT

Wordpress Report
Date: 2019-04-08

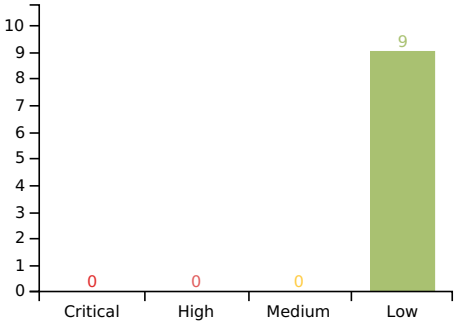
1. Executive Summary

Project Name:	Wordpress Report	Analyzed Files:	715
Analysis Start Date:	2019-04-06, 10:50	Analyzed LOC:	356,851
Analysis End Date:	2019-04-06, 10:59	Analyzed Issue Types:	210
Analysis Time:	8m 52s	Detected Issues:	9
Engine Version(s):	php-preparser 3.0.0 php-engine 3.0.4	Max Issues per Type:	500

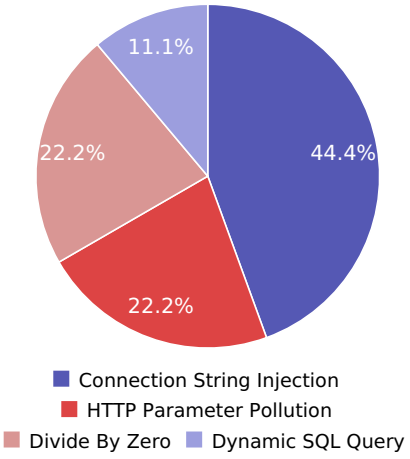
Risk Matrix



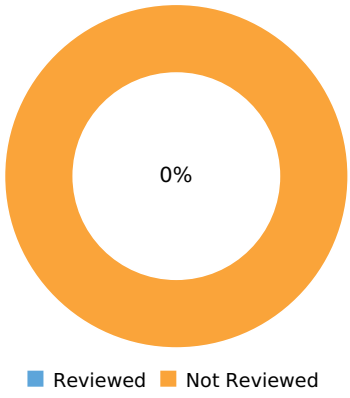
Vulnerabilities by Risk



Top Vulnerability Types



Review Status



2. Issue Breakdown

The detected security issues in this project are categorized as follows.

Severity	Vulnerability Type	CWE [?]	OWASP Top 10 [?]	SANS 25 [?]	PCI DSS [?]	ASVS [?]	Issues
Low	Connection String Injection	99	A2	Rank 16	6.5.4		4
Low	HTTP Parameter Pollution	233	A2	Not Ranked	6.5.4	5.1.1	2
Low	Divide By Zero	369		Not Ranked			2
Low	Dynamic SQL Query	89		Not Ranked			1

3. Issue Details

In the following, all security issues detected in the analyzed project are presented in detail. The issues are grouped by vulnerability type and by the detected markup context. A *markup context* represents the position of user-supplied data (*source*) used in a sensitive operation (*sink*). Depending on the markup context, an attacker can alter the operation and different security mechanisms must be applied in order to patch the security issue thoroughly.

3.1. Connection String Injection

OWASP Top 10:	2017: A2
CWE:	99
SANS 25:	Rank 16
PCI DSS:	6.5.4
Severity:	Low

A connection string injection vulnerability occurs when user input is used as a connection credential to a resource. An attacker can misuse this behavior by performing brute force attacks against the credentials or by tricking the server into connecting to their own resource that interacts with the web application in an unexpected way.

A connection string injection vulnerability occurs when user input is used as part of the connection details to a resource. If any part of the connection details needs to be dynamic, it is recommended to restrict it by a whitelist.

Issue #87349 - wordpress/wp-includes/Requests/Transport/fsockopen.php: 123

Path:	wordpress/wp-includes/Requests/Transport/fsockopen.php
Line:	123
Sink:	stream_socket_client
Source:	_POST
Taint:	HTTP

Code Summary

The POST parameter 'file' is received in line 83 of the file wordpress/wp-admin/plugin-editor.php.

The user-supplied data is concatenated into uri markup in line 117 of the file wordpress/wp-includes/Requests/Transport/fsockopen.php in the method Requests_Transport_fsockopen::request().

The user-supplied data is then used unsanitized in the sensitive operation stream_socket_client() in line 123 of the file wordpress/wp-includes/Requests/Transport/fsockopen.php in the method Requests_Transport_fsockopen::request(). Please refer to the context and description for further information.

wordpress/wp-admin/includes/file.php

```
352  function wp_edit_theme_plugin_file( $args ) {
⋮
356  $file = $args['file'];
⋮
370  $plugin = null;
⋮
403  $stylesheet = $args['theme'];
⋮
```

```
496  $scrape_key = md5( rand() );
⋮
498  $scrape_nonce = strval( rand() );
⋮
502  $scrape_params = array(
503  'wp_scrape_key' => $scrape_key,
504  'wp_scrape_nonce' => $scrape_nonce,
505  );
⋮
526  $url = add_query_arg( compact( 'plugin', 'file' ), admin_url( 'plugin-editor.php' ) );
⋮
528  $url = add_query_arg(
529  array(
530  'theme' => $stylesheet,
531  'file' => $file,
532  ),
533  admin_url( 'theme-editor.php' )
534  );
⋮
536  $url = admin_url();
⋮
538  $url = add_query_arg( $scrape_params, $url );
539  $r = wp_remote_get( $url, compact( 'cookies', 'headers', 'timeout' ) );
⋮
608  }
```

wordpress/wp-admin/plugin-editor.php

```
83  $r = wp_edit_theme_plugin_file( wp_unslash( $_POST ) );
```

wordpress/wp-includes/Requests/Transport/fsockopen.php

```
15   class Requests_Transport_fsockopen implements Requests_Transport {
⋮
58   public function request($url, $headers = array(), $data = array(), $options = array()) {
⋮
61   $url_parts = parse_url($url);
⋮
65   $host = $url_parts['host'];
⋮
72   $remote_socket = 'ssl://' . $host;
⋮
74   $url_parts['port'] = 443;
⋮
109  $remote_socket = 'tcp://' . $host;
⋮
115  $url_parts['port'] = 80;
⋮
117  $remote_socket .= ':' . $url_parts['port'];
⋮
123  $socket = stream_socket_client($remote_socket, $errno, $errstr, ceil($options['connect_timeout']), STREAM_CLIE
     NT_CONNECT, $context);
⋮
292  }
⋮
444  }
```

wordpress/wp-includes/class-http.php

```
28    class WP_Http {
⋮
149   public function request( $url, $args = array() ) {
⋮
265   $url = wp_http_validate_url( $url );
⋮
```

```
268   $url = wp_kses_bad_protocol( $url, array( 'http', 'https', 'ssl' ) );
⋮
384   $requests_response = Requests::request( $url, $headers, $data, $type, $options );
⋮
437   }
⋮
610   public function get( $url, $args = array() ) {
⋮
612   $r = wp_parse_args( $args, $defaults );
613   return $this->request( $url, $r );
614   }
⋮
1061  }
```

wordpress/wp-includes/class-requests.php

```
21    class Requests {
⋮
357   public static function request($url, $headers = array(), $data = array(), $type = self::GET, $options = array()) {
⋮
379   $response = $transport->request($url, $headers, $data, $options);
⋮
384   }
⋮
980   }
```

wordpress/wp-includes/http.php

```
168   function wp_remote_get( $url, $args = array() ) {
169   $http = _wp_http_get_object();
170   return $http->get( $url, $args );
171   }
```

URI Context

The following snippet(s) do not represent actual code but the tainted context.

```
tcp://P_576_:P_600_:P_600_:P_600_:U_1788_?'Array'=https://target/wp-admin/theme-editor.php?
theme=P_1790_&file= $_POST['file'] &
```

Issue #87412 - wordpress/wp-includes/Requests/Transport/fsockopen.php: 123

Path:	wordpress/wp-includes/Requests/Transport/fsockopen.php
Line:	123
Sink:	stream_socket_client
Source:	_POST
Taint:	HTTP

Code Summary

The POST parameter 'file' is received in line 117 of the file wordpress/wp-admin/theme-editor.php.

The user-supplied data is concatenated into uri markup in line 117 of the file wordpress/wp-includes/Requests/Transport/fsockopen.php in the method Requests_Transport_fsockopen::request().

The user-supplied data is then used unsanitized in the sensitive operation stream_socket_client() in line 123 of the file wordpress/wp-includes/Requests/Transport/fsockopen.php in the method Requests_Transport_fsockopen::request(). Please refer to the context and description for further information.

wordpress/wp-admin/includes/file.php

```
352 function wp_edit_theme_plugin_file( $args ) {
⋮
356 $file = $args['file'];
⋮
370 $plugin = null;
⋮
403 $stylesheet = $args['theme'];
⋮
496 $scrape_key = md5( rand() );
⋮
498 $scrape_nonce = strval( rand() );
⋮
502 $scrape_params = array(
503 'wp_scrape_key' => $scrape_key,
504 'wp_scrape_nonce' => $scrape_nonce,
505 );
⋮
526 $url = add_query_arg( compact( 'plugin', 'file' ), admin_url( 'plugin-editor.php' ) );
⋮
528 $url = add_query_arg(
529 array(
530 'theme' => $stylesheet,
531 'file' => $file,
532 ),
533 admin_url( 'theme-editor.php' )
534 );
⋮
536 $url = admin_url();
⋮
538 $url = add_query_arg( $scrape_params, $url );
539 $r = wp_remote_get( $url, compact( 'cookies', 'headers', 'timeout' ) );
⋮
608 }
```

wordpress/wp-admin/theme-editor.php

```
117 $r = wp_edit_theme_plugin_file( wp_unslash( $_POST ) );
```

wordpress/wp-includes/Requests/Transport/fsockopen.php

```
15 class Requests_Transport_fsockopen implements Requests_Transport {
⋮
58 public function request($url, $headers = array(), $data = array(), $options = array()) {
⋮
61 $url_parts = parse_url($url);
⋮
65 $host = $url_parts['host'];
⋮
72 $remote_socket = 'ssl://' . $host;
⋮
74 $url_parts['port'] = 443;
⋮
109 $remote_socket = 'tcp://' . $host;
⋮
115 $url_parts['port'] = 80;
⋮
117 $remote_socket .= ':' . $url_parts['port'];
⋮
123 $socket = stream_socket_client($remote_socket, $errno, $errstr, ceil($options['connect_timeout']), STREAM_CLIE
    NT_CONNECT, $context);
⋮
292 }
⋮
444 }
```

wordpress/wp-includes/class-http.php

```
28    class WP_Http {
⋮
149   public function request( $url, $args = array() ) {
⋮
265   $url = wp_http_validate_url( $url );
⋮
268   $url = wp_kses_bad_protocol( $url, array( 'http', 'https', 'ssl' ) );
⋮
384   $requests_response = Requests::request( $url, $headers, $data, $type, $options );
⋮
437   }
⋮
610   public function get( $url, $args = array() ) {
⋮
612   $r = wp_parse_args( $args, $defaults );
613   return $this->request( $url, $r );
614   }
⋮
1061  }
```

wordpress/wp-includes/class-requests.php

```
21    class Requests {
⋮
357   public static function request($url, $headers = array(), $data = array(), $type = self::GET, $options = array()) {
⋮
379   $response = $transport->request($url, $headers, $data, $options);
⋮
384   }
⋮
980   }
```

wordpress/wp-includes/http.php

```
168   function wp_remote_get( $url, $args = array() ) {
169   $http = _wp_http_get_object();
170   return $http->get( $url, $args );
171   }
```

URI Context

The following snippet(s) do not represent actual code but the tainted context.

```
tcp://P_576_:P_600_:P_600_:P_600_:U_1788_?'Array'=https://target/wp-admin/theme-editor.php?
theme=P_1790_&file= $_POST['file'] &
```

Issue #87560 - wordpress/wp-includes/Requests/Transport/fsockopen.php: 123

Path:	wordpress/wp-includes/Requests/Transport/fsockopen.php
Line:	123
Sink:	stream_socket_client
Source:	_POST
Taint:	HTTP

Code Summary

The POST parameter 'file' is received in line 4420 of the file wordpress/wp-admin/includes/ajax-actions.php in the function wp_ajax_edit_theme_plugin_file().

The user-supplied data is concatenated into uri markup in line 117 of the file wordpress/wp-

includes/Requests/Transport/fsockopen.php in the method Requests_Transport_fsockopen::request().

The user-supplied data is then used unsanitized in the sensitive operation stream_socket_client() in line 123 of the file wordpress/wp-includes/Requests/Transport/fsockopen.php in the method Requests_Transport_fsockopen::request(). Please refer to the context and description for further information.

wordpress/wp-admin/includes/ajax-actions.php

```
4419  function wp_ajax_edit_theme_plugin_file() {
4420  $r = wp_edit_theme_plugin_file( wp_unslash( $_POST ) ); // Validation of args is done in wp_edit_theme_plugin_file().
⋮
4438  }
```

wordpress/wp-admin/includes/file.php

```
352  function wp_edit_theme_plugin_file( $args ) {
⋮
356  $file = $args['file'];
⋮
370  $plugin = null;
⋮
403  $stylesheet = $args['theme'];
⋮
496  $scrape_key = md5( rand() );
⋮
498  $scrape_nonce = strval( rand() );
⋮
502  $scrape_params = array(
503  'wp_scrape_key' => $scrape_key,
504  'wp_scrape_nonce' => $scrape_nonce,
505  );
⋮
526  $url = add_query_arg( compact( 'plugin', 'file' ), admin_url( 'plugin-editor.php' ) );
⋮
528  $url = add_query_arg(
529  array(
530  'theme' => $stylesheet,
531  'file' => $file,
532  ),
533  admin_url( 'theme-editor.php' )
534  );
⋮
536  $url = admin_url();
⋮
538  $url = add_query_arg( $scrape_params, $url );
539  $r = wp_remote_get( $url, compact( 'cookies', 'headers', 'timeout' ) );
⋮
608  }
```

wordpress/wp-includes/Requests/Transport/fsockopen.php

```
15  class Requests_Transport_fsockopen implements Requests_Transport {
⋮
58  public function request($url, $headers = array(), $data = array(), $options = array()) {
⋮
61  $url_parts = parse_url($url);
⋮
65  $host = $url_parts['host'];
⋮
72  $remote_socket = 'ssl://' . $host;
```

```
⋮
74   $url_parts['port'] = 443;
⋮
109  $remote_socket = 'tcp://' . $host;
⋮
115  $url_parts['port'] = 80;
⋮
117  $remote_socket .= ':' . $url_parts['port'];
⋮
123  $socket = stream_socket_client($remote_socket, $errno, $errstr, ceil($options['connect_timeout']), STREAM_CLIE
     NT_CONNECT, $context);
⋮
292  }
⋮
444  }
```

wordpress/wp-includes/class-http.php

```
28    class WP_Http {
⋮
149   public function request( $url, $args = array() ) {
⋮
265   $url = wp_http_validate_url( $url );
⋮
268   $url = wp_kses_bad_protocol( $url, array( 'http', 'https', 'ssl' ) );
⋮
384   $requests_response = Requests::request( $url, $headers, $data, $type, $options );
⋮
437   }
⋮
610   public function get( $url, $args = array() ) {
⋮
612   $r = wp_parse_args( $args, $defaults );
613   return $this->request( $url, $r );
614   }
⋮
1061  }
```

wordpress/wp-includes/class-requests.php

```
21   class Requests {
⋮
357  public static function request($url, $headers = array(), $data = array(), $type = self::GET, $options = array()) {
⋮
379  $response = $transport->request($url, $headers, $data, $options);
⋮
384  }
⋮
980  }
```

wordpress/wp-includes/http.php

```
168  function wp_remote_get( $url, $args = array() ) {
169  $http = _wp_http_get_object();
170  return $http->get( $url, $args );
171  }
```

URI Context

The following snippet(s) do not represent actual code but the tainted context.

```
tcp://P_576_:P_600_:P_600_:P_600_:U_1788_?'Array'=https://target/wp-admin/theme-editor.php?
theme=P_1790_&file= $_POST['file'] &
```

Issue #87617 - wordpress/wp-includes/Requests/Transport/fsockopen.php: 123

Path:	wordpress/wp-includes/Requests/Transport/fsockopen.php
Line:	123
Sink:	stream_socket_client
Source:	_POST
Taint:	HTTP

Code Summary

The POST parameter '_wp_attached_file' is received in line 5610 of the file wordpress/wp-includes/post.php in the function wp_get_attachment_url().

The user-supplied data is concatenated into uri markup in line 117 of the file wordpress/wp-includes/Requests/Transport/fsockopen.php in the method Requests_Transport_fsockopen::request().

The user-supplied data is then used unsanitized in the sensitive operation stream_socket_client() in line 123 of the file wordpress/wp-includes/Requests/Transport/fsockopen.php in the method Requests_Transport_fsockopen::request(). Please refer to the context and description for further information.

wordpress/wp-includes/Requests/Transport/fsockopen.php

```
15   class Requests_Transport_fsockopen implements Requests_Transport {
⋮
58   public function request($url, $headers = array(), $data = array(), $options = array()) {
⋮
61   $url_parts = parse_url($url);
⋮
65   $host = $url_parts['host'];
⋮
72   $remote_socket = 'ssl://' . $host;
⋮
74   $url_parts['port'] = 443;
⋮
109  $remote_socket = 'tcp://' . $host;
⋮
115  $url_parts['port'] = 80;
⋮
117  $remote_socket .= ':' . $url_parts['port'];
⋮
123  $socket = stream_socket_client($remote_socket, $errno, $errstr, ceil($options['connect_timeout']), STREAM_CLIE
     NT_CONNECT, $context);
⋮
292  }
⋮
444  }
```

wordpress/wp-includes/class-http.php

```
28    class WP_Http {
⋮
149   public function request( $url, $args = array() ) {
⋮
265   $url = wp_http_validate_url( $url );
⋮
268   $url = wp_kses_bad_protocol( $url, array( 'http', 'https', 'ssl' ) );
⋮
384   $requests_response = Requests::request( $url, $headers, $data, $type, $options );
⋮
437   }
```

```
⋮
610   public function get( $url, $args = array() ) {
⋮
612   $r = wp_parse_args( $args, $defaults );
613   return $this->request( $url, $r );
614   }
⋮
1061  }
```

wordpress/wp-includes/class-oembed.php

```
19    class WP_oEmbed {
⋮
254   public function get_provider( $url, $args = '' ) {
⋮
279   $provider = $this->discover( $url );
⋮
283   }
⋮
345   public function get_data( $url, $args = '' ) {
346   $args = wp_parse_args( $args );
⋮
348   $provider = $this->get_provider( $url, $args );
⋮
361   }
⋮
375   public function get_html( $url, $args = '' ) {
⋮
397   $data = $this->get_data( $url, $args );
⋮
413   }
⋮
423   public function discover( $url ) {
⋮
425   $args = array(
426   'limit_response_size' => 153600, // 150 KB
427   );
⋮
439   $args = apply_filters( 'oembed_remote_get_args', $args, $url );
⋮
442   $request = wp_safe_remote_get( $url, $args );
⋮
501   }
⋮
747   }
```

wordpress/wp-includes/class-requests.php

```
21    class Requests {
⋮
357   public static function request($url, $headers = array(), $data = array(), $type = self::GET, $options = array()) {
⋮
379   $response = $transport->request($url, $headers, $data, $options);
⋮
384   }
⋮
980   }
```

wordpress/wp-includes/embed.php

```
98    function wp_oembed_get( $url, $args = '' ) {
99    $oembed = _wp_oembed_get_object();
100   return $oembed->get_html( $url, $args );
101   }
```

wordpress/wp-includes/http.php

```
67  function wp_safe_remote_get( $url, $args = array() ) {
68  $args['reject_unsafe_urls'] = true;
69  $http = _wp_http_get_object();
70  return $http->get( $url, $args );
71  }
```

wordpress/wp-includes/widgets/class-wp-widget-media-video.php

```
17    class WP_Widget_Media_Video extends WP_Widget_Media {
⋮
114   public function render_media( $instance ) {
115   $instance = array_merge( wp_list_pluck( $this->get_instance_schema(), 'default' ), $instance );
⋮
122   $src = $instance['url'];
⋮
124   $src = wp_get_attachment_url( $attachment->ID );
⋮
147   echo $this->inject_video_max_width_style( wp_oembed_get( $src ) );
⋮
149   }
⋮
258   }
```

URI Context

The following snippet(s) do not represent actual code but the tainted context.

```
tcp://P_576_:P_600_:P_600_:P_600_:http: UPLOADS/sites/123
```

3.1. HTTP Parameter Pollution

ASVS:	4.0.1: 5.1.1
OWASP Top 10:	2017: A2
CWE:	233
PCI DSS:	6.5.4
Severity:	Low

An HTTP Parameter Pollution (HPP) vulnerability occurs when unsanitized user input is used to construct a URL and its query parameters. An attacker can modify the URL and insert additional query string parameters that could overwrite existing ones and thereby change the intended behavior of the request.

To prevent HTTP Parameter Pollution attacks, it is recommended to urlencode() all values that are embedded into query string parameters such that no additional parameters can be added.

Issue #87359 - wordpress/wp-includes/Requests/Transport/cURL.php: 375

Path:	wordpress/wp-includes/Requests/Transport/cURL.php
Line:	375
Sink:	curl_setopt
Source:	_POST
Taint:	HTTP

Code Summary

The POST parameter '_wp_attached_file' is received in line 453 of the file wordpress/wp-includes/post.php in the function get_attached_file().

The user-supplied data is then used unsanitized in the sensitive operation curl_setopt() in line 375 of the file wordpress/wp-includes/Requests/Transport/cURL.php in the method Requests_Transport_cURL::setup_handle(). Please refer to the context and description for further information.

wordpress/wp-admin/includes/class-plugin-upgrader.php

```
21   class Plugin_Upgrader extends WP_Upgrader {
⋮
90   public function install( $package, $args = array() ) {
⋮
106  $this->run(
107  array(
108  'package' => $package,
109  'destination' => WP_PLUGIN_DIR,
110  'clear_destination' => false, // Do not overwrite files.
111  'clear_working' => true,
112  'hook_extra' => array(
113  'type' => 'plugin',
114  'action' => 'install',
115  ),
116  )
117  );
⋮
130  }
⋮
487  }
```

wordpress/wp-admin/includes/class-wp-upgrader.php

```
51   class WP_Upgrader {
⋮
251  public function download_package( $package ) {
⋮
278  $download_file = download_url( $package );
⋮
285  }
⋮
658  public function run( $options ) {
⋮
670  $options = wp_parse_args( $options, $defaults );
⋮
702  $options = apply_filters( 'upgrader_package_options', $options );
⋮
733  $download = $this->download_package( $options['package'] );
⋮
817  }
⋮
904  }
```

wordpress/wp-admin/includes/file.php

```
973   function download_url( $url, $timeout = 300 ) {

⋮
986   $response = wp_safe_remote_get(
987   $url,
988   array(
989   'timeout' => $timeout,
990   'stream' => true,
991   'filename' => $tmpfname,
992   )
993   );
⋮
1038  }
```

wordpress/wp-admin/update.php

```
149  $file_upload = new File_Upload_Upgrader( 'pluginzip', 'package' );
⋮
161  $upgrader = new Plugin_Upgrader( new Plugin_Installer_Skin( compact( 'type', 'title', 'nonce', 'url' ) ) );
162  $result = $upgrader->install( $file_upload->package );
```

wordpress/wp-includes/Requests/Transport/cURL.php

```
15   class Requests_Transport_cURL implements Requests_Transport {
⋮
130  public function request($url, $headers = array(), $data = array(), $options = array()) {
131  $this->hooks = $options['hooks'];
⋮
133  $this->setup_handle($url, $headers, $data, $options);
⋮
185  }
⋮
309  protected function setup_handle($url, $headers, $data, $options) {
⋮
323  $url = self::format_get($url, $data);
⋮
375  curl_setopt($this->handle, CURLOPT_URL, $url);
⋮
393  }
⋮
542  }
```

wordpress/wp-includes/class-http.php

```
28    class WP_Http {
⋮
149   public function request( $url, $args = array() ) {
⋮
265   $url = wp_http_validate_url( $url );
⋮
268   $url = wp_kses_bad_protocol( $url, array( 'http', 'https', 'ssl' ) );
⋮
384   $requests_response = Requests::request( $url, $headers, $data, $type, $options );
⋮
437   }
⋮
610   public function get( $url, $args = array() ) {
⋮
612   $r = wp_parse_args( $args, $defaults );
613   return $this->request( $url, $r );
614   }
⋮
1061  }
```

wordpress/wp-includes/class-requests.php

```
21   class Requests {
⋮
357  public static function request($url, $headers = array(), $data = array(), $type = self::GET, $options = array()) {
⋮
379  $response = $transport->request($url, $headers, $data, $options);
⋮
384  }
⋮
980  }
```

wordpress/wp-includes/http.php

```
67  function wp_safe_remote_get( $url, $args = array() ) {
68  $args['reject_unsafe_urls'] = true;
```

```
69  $http = _wp_http_get_object();
70  return $http->get( $url, $args );
71  }
```

Parameter Context

The following snippet(s) do not represent actual code but the tainted context.

```
/sites/123/Y-m-d H:i:s/Y-m-d H:i:s/wordpress/UPLOADS/sites/123/Y-m-d H:i:s/Y-m-d H:i:sfalse
```

Issue #87616 - wordpress/wp-includes/Requests/Transport/cURL.php: 375

Path:	wordpress/wp-includes/Requests/Transport/cURL.php
Line:	375
Sink:	curl_setopt
Source:	_POST
Taint:	HTTP

Code Summary

The POST parameter '_wp_attached_file' is received in line 5610 of the file wordpress/wp-includes/post.php in the function wp_get_attachment_url().

The user-supplied data is concatenated into parameter markup in line 613 of the file wordpress/wp-includes/class-http.php in the method WP_Http::get().

The user-supplied data is then used unsanitized in the sensitive operation curl_setopt() in line 375 of the file wordpress/wp-includes/Requests/Transport/cURL.php in the method Requests_Transport_cURL::setup_handle(). Please refer to the context and description for further information.

wordpress/wp-includes/Requests/Transport/cURL.php

```
15   class Requests_Transport_cURL implements Requests_Transport {
⋮
130  public function request($url, $headers = array(), $data = array(), $options = array()) {
131  $this->hooks = $options['hooks'];
⋮
133  $this->setup_handle($url, $headers, $data, $options);
⋮
185  }
⋮
309  protected function setup_handle($url, $headers, $data, $options) {
⋮
323  $url = self::format_get($url, $data);
⋮
375  curl_setopt($this->handle, CURLOPT_URL, $url);
⋮
393  }
⋮
542  }
```

wordpress/wp-includes/class-http.php

```
28    class WP_Http {
⋮
149   public function request( $url, $args = array() ) {
⋮
265   $url = wp_http_validate_url( $url );
⋮
```

```
268   $url = wp_kses_bad_protocol( $url, array( 'http', 'https', 'ssl' ) );
⋮
384   $requests_response = Requests::request( $url, $headers, $data, $type, $options );
⋮
437   }
⋮
610   public function get( $url, $args = array() ) {
⋮
612   $r = wp_parse_args( $args, $defaults );
613   return $this->request( $url, $r );
614   }
⋮
1061  }
```

wordpress/wp-includes/class-oembed.php

```
19    class WP_oEmbed {
⋮
254   public function get_provider( $url, $args = '' ) {
⋮
279   $provider = $this->discover( $url );
⋮
283   }
⋮
345   public function get_data( $url, $args = '' ) {
346   $args = wp_parse_args( $args );
⋮
348   $provider = $this->get_provider( $url, $args );
⋮
361   }
⋮
375   public function get_html( $url, $args = '' ) {
⋮
397   $data = $this->get_data( $url, $args );
⋮
413   }
⋮
423   public function discover( $url ) {
⋮
425   $args = array(
426   'limit_response_size' => 153600, // 150 KB
427   );
⋮
439   $args = apply_filters( 'oembed_remote_get_args', $args, $url );
⋮
442   $request = wp_safe_remote_get( $url, $args );
⋮
501   }
⋮
747   }
```

wordpress/wp-includes/class-requests.php

```
21    class Requests {
⋮
357   public static function request($url, $headers = array(), $data = array(), $type = self::GET, $options = array()) {
⋮
379   $response = $transport->request($url, $headers, $data, $options);
⋮
384   }
⋮
980   }
```

wordpress/wp-includes/embed.php

```
98  function wp_oembed_get( $url, $args = '' ) {
99  $oembed = _wp_oembed_get_object();
100 return $oembed->get_html( $url, $args );
101 }
```

wordpress/wp-includes/http.php

```
67  function wp_safe_remote_get( $url, $args = array() ) {
68  $args['reject_unsafe_urls'] = true;
69  $http = _wp_http_get_object();
70  return $http->get( $url, $args );
71  }
```

wordpress/wp-includes/widgets/class-wp-widget-media-video.php

```
17  class WP_Widget_Media_Video extends WP_Widget_Media {
⋮
114 public function render_media( $instance ) {
115 $instance = array_merge( wp_list_pluck( $this->get_instance_schema(), 'default' ), $instance );
⋮
122 $src = $instance['url'];
⋮
124 $src = wp_get_attachment_url( $attachment->ID );
⋮
147 echo $this->inject_video_max_width_style( wp_oembed_get( $src ) );
⋮
149 }
⋮
258 }
```

Parameter Context

The following snippet(s) do not represent actual code but the tainted context.

```
P_600_:P_600_:P_600_:http: UPLOADS/sites/123
```

3.1. Divide By Zero

CWE: 369
Severity: Low

The application potentially divides a value by zero. This can be abused by an attacker to provoke errors that could leave the application in an undefined behavior.

To resolve this issue you can check if the divisor is zero before using it.

Issue #87100 - wordpress/wp-includes/theme-compat/embed-content.php: 42

Path: wordpress/wp-includes/theme-compat/embed-content.php
Line: 42
Sink:
Source: _POST
Taint: HTTP

Code Summary

The POST parameter 'file_metadata[sizes][0][height]' is received in line 39 of the file wordpress/wp-includes/theme-compat/embed-content.php.

A code quality issue was detected in line 42 of the file wordpress/wp-includes/theme-compat/embed-content.php. Please refer to the context and description for further information.

wordpress/wp-includes/theme-compat/embed-content.php

```
39  $meta = wp_get_attachment_metadata( $thumbnail_id );
⋮
41  foreach ( $meta['sizes'] as $size => $data ) {
42  if ( $data['height'] > 0 && $data['width'] / $data['height'] > $aspect_ratio ) {
43  $aspect_ratio = $data['width'] / $data['height'];
44  $measurements = array( $data['width'], $data['height'] );
45  $image_size = $size;
46  }
47  }
```

Math Context

The following snippet(s) do not represent actual code but the tainted context.

```
$_POST['file_metadata']['sizes'][$data[.*]]['height']
```

Issue #87515 - wordpress/wp-admin/includes/image.php: 320

Path:	wordpress/wp-admin/includes/image.php
Line:	320
Sink:	
Source:	_POST
Taint:	Multi-Step

Code Summary

A file upload allows to bypass a validation of the value '_wp_attached_file' that is received in line 453 of the file wordpress/wp-includes/post.php in the function get_attached_file().

A code quality issue was detected in line 320 of the file wordpress/wp-admin/includes/image.php in the function wp_exif_frac2dec(). Please refer to the context and description for further information.

wordpress/wp-admin/custom-header.php

```
14    class Custom_Image_Header {
⋮
758   public function step_2() {
⋮
770   $file = get_attached_file( $attachment_id, true );
⋮
774   $data = $this->step_2_manage_upload();
⋮
776   $file = $data['file'];
⋮
805   wp_update_attachment_metadata( $attachment_id, wp_generate_attachment_metadata( $attachment_id, $file
      ) );
⋮
872   }
⋮
1502  }
```

wordpress/wp-admin/includes/image.php

```
317  function wp_exif_frac2dec( $str ) {
⋮
```

```
320  return $n / $d;
⋮
323  }
```

Math Context

The following snippet(s) do not represent actual code but the tainted context.

```
wordpress/UPLOADS/sites/123/ $_POST['_wp_attached_file']
```

3.1. Dynamic SQL Query

CWE: 89
Severity: Low

A SQL query is constructed dynamically by concatenation. This can lead to SQL injection attacks.

It is recommended to use prepared statements for all SQL queries. The prepared statement itself should only use placeholders for data and never concatenate data directly into the query.

Issue #87277 - wordpress/wp-includes/post.php: 5476

Path: wordpress/wp-includes/post.php
Line: 5476
Sink: execute
Source: _POST
Taint: HTTP

Code Summary

A code quality issue was detected in line 5476 of the file wordpress/wp-includes/post.php in the function wp_delete_attachment_files(). Please refer to the context and description for further information.

wordpress/wp-includes/post.php

```
5382  function wp_delete_attachment( $post_id, $force_delete = false ) {
⋮
5404  $meta = wp_get_attachment_metadata( $post_id );
⋮
5450  wp_delete_attachment_files( $post_id, $meta, $backup_sizes, $file );
⋮
5455  }
⋮
5468  function wp_delete_attachment_files( $post_id, $meta, $backup_sizes, $file ) {
⋮
5476  if ( ! $wpdb->get_row( $wpdb->prepare( "SELECT meta_id FROM $wpdb->postmeta WHERE meta_key = '_wp_attachment_metadata' AND meta_value LIKE %s AND post_id <> %d", '%' . $wpdb->esc_like( $meta['thumb'] ) . '%', $post_id ) ) ) {
⋮
5523  }
```

SQL Context

The following snippet(s) do not represent actual code but the tainted context.

```
SELECT meta_id FROM WHERE meta_key = '_wp_attachment_metadata' AND meta_value LIKE %
$_POST['file_metadata']['thumb'] % AND post_id <> 1
```