â\224\214â\224\200â\224\200â\224\200â\224\200â\224\200â\224\200â\224\200â\224\200â\224\200â\224\200â
\224\200â\224\200â\224\200â\224\200â\224\200â\224\200â\224\200â\224\200â\224\200â\224\220
â\224\202 24 Code Findings â\224\202
â\224\224â\224\200â\224\200â\224\200â\224\200â\224\200â\224\200â\224\200â\224\200â\224\200â\224\200â
\224\200â\224\200â\224\200â\224\200â\224\200â\224\200â\224\200â\224\200â\224\200â\224\230

\033[36m\033[22m\033[24m /home/kali/Downloads/really-simple-ssl/lets-encrypt/class-letsencr
ypt-handler.php\033[0m
â\235¯â\235± \033[1mlang.security.unlink-use\033[0m
Using user input when deleting files with `unlink()` is potentially dangerous. A
malicious actor could use this to modify or access files they have no right to.

1534â\224\206 unlink( $file );

\033[36m\033[22m\033[24m /home/kali/Downloads/really-simple-ssl/lets-encrypt/integrations/c
panel/cpanel.php\033[0m
â\235¯â\235¯â\235± \033[1mlang.security.curl-ssl-verifypeer-off\033[0m
SSL verification is disabled but should not be (currently CURLOPT_SSL_VERIFYPEER=
false)

227â\224\206 curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
â\213®â\224\206----------------------------------------
â\235¯â\235¯â\235± \033[1mlang.security.curl-ssl-verifypeer-off\033[0m
SSL verification is disabled but should not be (currently CURLOPT_SSL_VERIFYPEER= 0)

277â\224\206 curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, 0);

\033[36m\033[22m\033[24m

/home/kali/Downloads/really-simple-ssl/lets-encrypt/integrations/directadmin/httpsocket.php
\033[0m
â\235¯â\235¯â\235± \033[1mlang.security.curl-ssl-verifypeer-off\033[0m
SSL verification is disabled but should not be (currently CURLOPT_SSL_VERIFYPEER=
false)

205â\224\206 curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false); //1

\033[36m\033[22m\033[24m /home/kali/Downloads/really-simple-ssl/lib/admin/class-encryption.
php\033[0m
â\235¯â\235± \033[1mlang.security.audit.openssl-decrypt-validate\033[0m
The function `openssl_decrypt` returns either a string of the decrypted data on

success or `false` on failure. If the failure case is not handled, this could lead t
o
undefined behavior in your application. Please handle the case where

`openssl_decrypt` returns `false`.

117â\224\206 $decrypted_data = openssl_decrypt( $encrypted_data, 'aes-256-cbc', $key
, 0,
$iv );

â\235¯â\235± \033[1mlang.security.unserialize-use\033[0m
Calling `unserialize()` with user input in the pattern can lead to arbitrary code
execution. Consider using JSON or structured data approaches (e.g. Google Protocol
Buffers).

```
          125â\224\206 $unserialized_data = @unserialize( $decrypted_data );

  \033[36m\033[22m\033[24m  /home/kali/Downloads/really-simple-ssl/security/firewall-manager.p
hp\033[0m
    â\235¯â\235± \033[1mlang.security.unlink-use\033[0m
          Using user input when deleting files with 'unlink()' is potentially dangerous. A
          malicious actor could use this to modify or access files they have no right to.

          241â\224\206 unlink( $this->file );//phpcs:ignore

  \033[36m\033[22m\033[24m  /home/kali/Downloads/really-simple-ssl/security/tests.php\033[0m
   â\235¯â\235¯â\235± \033[1mlang.security.curl-ssl-verifypeer-off\033[0m
          SSL verification is disabled but should not be (currently CURLOPT_SSL_VERIFYPEER=
          false)

          104â\224\206 curl_setopt( $ch, CURLOPT_SSL_VERIFYPEER, false );

    â\235¯â\235± \033[1mlang.security.unlink-use\033[0m
          Using user input when deleting files with 'unlink()' is potentially dangerous. A
          malicious actor could use this to modify or access files they have no right to.

          352â\224\206 unlink($test_file);


  \033[36m\033[22m\033[24m  /home/kali/Downloads/really-simple-ssl/security/wordpress/hide-wp-
version.php\033[0m
   â\235¯â\235¯â\235± \033[1mlang.security.weak-crypto\033[0m
          Detected usage of weak crypto function. Consider using stronger alternatives.

           54â\224\206 $this->new_version = hash( 'md5', $token );


  \033[36m\033[22m\033[24m  /home/kali/Downloads/really-simple-ssl/security/wordpress/rename-a
dmin-user.php\033[0m
    â\235¯â\235± \033[1mlang.security.unserialize-use\033[0m
          Calling 'unserialize()' with user input in the pattern can lead to arbitrary code
          execution. Consider using JSON or structured data approaches (e.g. Google Protocol
          Buffers).

          143â\224\206 $unserialized = unserialize( $site_admins );


  \033[36m\033[22m\033[24m  /home/kali/Downloads/really-simple-ssl/security/wordpress/vulnerab
ilities.php\033[0m
   â\235¯â\235¯â\235± \033[1mlang.security.weak-crypto\033[0m
          Detected usage of weak crypto function. Consider using stronger alternatives.

          1631â\224\206 $random_string = md5( time() );


  \033[36m\033[22m\033[24m

  /home/kali/Downloads/really-simple-ssl/security/wordpress/vulnerabilities/FileStorage.php
\033[0m
   â\235¯â\235¯â\235± \033[1mlang.security.weak-crypto\033[0m
          Detected usage of weak crypto function. Consider using stronger alternatives.

           85â\224\206 $this->hash = md5(uniqid(rand(), true));

    â\235¯â\235± \033[1mlang.security.unlink-use\033[0m
          Using user input when deleting files with 'unlink()' is potentially dangerous. A
          malicious actor could use this to modify or access files they have no right to.
```

```
          112â\224\206 unlink($file);


  \033[36m\033[22m\033[24m

  /home/kali/Downloads/really-simple-ssl/security/wordpress/vulnerabilities/class-rsssl-file-s
torage.php\033[
  0m

    â\235¯â\235± \033[1mlang.security.unlink-use\033[0m
          Using user input when deleting files with `unlink()` is potentially dangerous. A
          malicious actor could use this to modify or access files they have no right to.

          120â\224\206 unlink( $file );
            â\213®â\224\206----------------------------------------
          150â\224\206 if (!unlink($dir.$obj))
            â\213®â\224\206----------------------------------------
          178â\224\206 unlink( $file );


  \033[36m\033[22m\033[24m

  /home/kali/Downloads/really-simple-ssl/security/wordpress/vulnerabilities/class-rsssl-folder
-name.php\033[0
  m

   â\235¯â\235¯â\235± \033[1mlang.security.weak-crypto\033[0m
          Detected usage of weak crypto function. Consider using stronger alternatives.

           21â\224\206 $newFolderName    = 'really-simple-ssl/' . md5( uniqid( mt_rand(), true
 ) );

  \033[36m\033[22m\033[24m  /home/kali/Downloads/really-simple-ssl/settings/settings.php\033[0
m
   â\235¯â\235¯â\235± \033[1mlang.security.injection.echoed-request\033[0m
          `Echo`ing user input risks cross-site scripting vulnerability. You should use
          `htmlentities()` when showing data to users.

           \033[32mâ\226¶â\226¶â\224\206 Autofix â\226¶ \033[0mecho htmlentities(json_encode($
response));
          303â\224\206 echo json_encode($response);

  \033[36m\033[22m\033[24m  /home/kali/Downloads/really-simple-ssl/upgrade.php\033[0m
    â\235¯â\235± \033[1mlang.security.unlink-use\033[0m
          Using user input when deleting files with `unlink()` is potentially dangerous. A
          malicious actor could use this to modify or access files they have no right to.

          150â\224\206 unlink( trailingslashit( WPMU_PLUGIN_DIR ) . 'rsssl_rest_api_optimizer.
php'
               );

  \033[36m\033[22m\033[24m  /home/kali/Downloads/really-simple-ssl/upgrade/upgrade-to-pro.php
\033[0m
   â\235¯â\235¯â\235± \033[1mlang.security.weak-crypto\033[0m
          Detected usage of weak crypto function. Consider using stronger alternatives.

          255â\224\206 $store_hash = md5( $this->api_url );

    â\235¯â\235± \033[1mlang.security.unlink-use\033[0m
          Using user input when deleting files with `unlink()` is potentially dangerous. A
          malicious actor could use this to modify or access files they have no right to.
```

```
        442â\224\206 unlink( trailingslashit( $new_dir ) . 'uninstall.php' );

  â\235¯â\235¯â\235± \033[1mlang.security.injection.echoed-request\033[0m
        `Echo`ing user input risks cross-site scripting vulnerability. You should use
        `htmlentities()` when showing data to users.

         \033[32mâ\226¶â\226¶â\224\206 Autofix â\226¶ \033[0mecho htmlentities($response);
        504â\224\206 echo $response;
          â\213®â\224\206----------------------------------------
         \033[32mâ\226¶â\226¶â\224\206 Autofix â\226¶ \033[0mecho htmlentities($response);
        697â\224\206 echo $response;
```