WWBN / AVideo Public

Code Issues 13 Pull requests Actions Projects Wiki Security a

Commit 1e6cf03

Daniel Neto committed on Mar 20 · 11 / 11

```
fix: Escape shell arguments in wget command to prevent command injection

https://github.com/WWBN/AVideo/security/advisories/GHSA-3fpm-8rjr-v5mc#event-587978
```

master · 29.0

1 parent dc3c825 commit 1e6cf03

1 file changed +160 -154 lines changed

Top

Filter files...

plugin/Live

test.php

1 file changed +160 -154 lines changed

Search within code

plugin/Live/test.php

```
@@ -1,154 +1,160 @@
- <?php
-
- $timeStarted = microtime(true);
-
- $statsURL = $_REQUEST['statsURL'];
- if (empty($statsURL) || $statsURL == "php://input" || !preg_match("/^http/",
  $statsURL)) {
-     _log('this is not a URL ');
-     exit;
- }
-
- ini_set('display_errors', 1);
```

```
- ini_set('display_startup_errors', 1);
- error_reporting(E_ALL & ~E_DEPRECATED);
-
- _log('Starting try to get URL ' . $statsURL);
-
- $result = url_get_contents($statsURL, 2);
-
- if ($result) {
-     _log('<span style="background-color: green; padding: 1px 4px; color: #FFF;">SUCCESS</span>');
- } else {
-     _log('<span style="background-color: red; padding: 1px 4px; color: #FFF;">FAIL</span>');
- }
-
- _log('Finish try to get URL ');
-
- $timeElapsed = number_format(microtime(true) - $timeStarted, 5);
- if ($timeElapsed>=2) {
-     _log('IMPORTANT: your stats took longer than 2 seconds to respond, the Streamer has a 2 seconds timeout rule ');
- }
-
- function url_get_contents($url, $timeout = 0)
- {
-     _log('url_get_contents start timeout=' . $timeout);
-     $agent = "AVideoStreamer";
-
-     $opts = [
-         'http' => ['header' => "User-Agent: {$agent}\r\n"],
-         "ssl" => [
-             "verify_peer" => false,
-             "verify_peer_name" => false,
-             "allow_self_signed" => true,
-         ],
-     ];
-     if (!empty($timeout)) {
-         ini_set('default_socket_timeout', $timeout);
-         $opts['http'] = ['timeout' => $timeout];
-     }
```

```
-
-    $context = stream_context_create($opts);
-
-    if (ini_get('allow_url_fopen')) {
-        try {
-            $tmp = file_get_contents($url, false, $context);
-            _log('file_get_contents:: '.htmlentities($tmp));
-            if (empty($tmp)) {
-                _log('file_get_contents fail return an empty content');
-                return false;
-            } else {
-                _log('file_get_contents works');
-                return true;
-            }
-        } catch (ErrorException $e) {
-            _log('file_get_contents fail catch error: ' . $e->getMessage());
-            return false;
-        }
-    } elseif (function_exists('curl_init')) {
-        _log('allow_url_fopen is NOT enabled but curl_init is, we will try CURL');
-        $ch = curl_init();
-        curl_setopt($ch, CURLOPT_USERAGENT, $agent);
-        curl_setopt($ch, CURLOPT_URL, $url);
-        curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
-        curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
-        curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);
-        if (!empty($timeout)) {
-            curl_setopt($ch, CURLOPT_TIMEOUT, $timeout);
-            curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, $timeout + 10);
-        }
-        $output = curl_exec($ch);
-        curl_close($ch);
-        _log('curl_init:: '.htmlentities($output));
-        if (empty($output)) {
-            _log('curl_init fail to download');
-            return false;
-        } else {
-            _log('curl_init success to download');
-            return true;
```

```
88  -         }
89  -     } else {
90  -         _log('IMPORTANT: allow_url_fopen is NOT enabled also curl_init is NOT
      enable, please investigate it and make sure it is enabled');
91  -     }
92  -
93  -
94  -     _log('Try wget');
95  -     // try wget
96  -     $tmpDir = sys_get_temp_dir();
97  -     if (empty($tmpDir)) {
98  -         _log('IMPORTANT: your sys_get_temp_dir is empty');
99  -         return false;
100 -     }
101 -     if (!is_writable($tmpDir)) {
102 -         _log('IMPORTANT: we cannot write in your temp directory ' . $tmpDir);
103 -         return false;
104 -     }
105 -     $tmpDir = rtrim($tmpDir, DIRECTORY_SEPARATOR) . DIRECTORY_SEPARATOR;
106 -
107 -     $filename = $tmpDir . md5($url);
108 -     if (wget($url, $filename)) {
109 -         $result = file_get_contents($filename);
110 -         _log('wget:: '.htmlentities($result));
111 -         unlink($filename);
112 -         if (!empty($result)) {
113 -             _log('wget works ');
114 -             return true;
115 -         } else {
116 -             _log('wget fail ');
117 -         }
118 -     }
119 -     unlink($filename);
120 -
121 -     return false;
122 - }
123 -
124 - function wget($url, $filename)
125 - {
126 -     if (empty($url) || $url == "php://input" || !preg_match("/^http/", $url)) {
```

```
-         _log('this is not a URL ');
-         return false;
-     }
-     if (strtoupper(substr(PHP_OS, 0, 3)) === 'WIN') {
-         _log('this is a windows OS ');
-         return false;
-     }
-     $cmd = "wget --tries=1 {$url} -O {$filename} --no-check-certificate";
-     exec($cmd);
-     if (!file_exists($filename)) {
-         _log('wget download fail, we cannot read the file: ' . $filename);
-         return false;
-     }
-     if (empty(filesize($filename))) {
-         _log('wget download fail, the file is empty: ' . $filename);
-         return false;
-     } else {
-         _log('wget download success, the file is NOT empty: ' . $filename);
-         return true;
-     }
- }
-
- function _log($msg)
- {
-     global $timeStarted;
-     $timeElapsed = number_format(microtime(true) - $timeStarted, 5);
-     echo '[' . date('Y-m-d H:i:s') . "] Time Elapsed: {$timeElapsed} seconds -
  " . $msg . '<br>' . PHP_EOL;
- }
+ <?php
+ require_once dirname(__FILE__) . '/../../videos/configuration.php';
+
+ if (!User::isAdmin()) {
+     http_response_code(403);
+     exit('Forbidden');
+ }
+
+ $timeStarted = microtime(true);
+
+ $statsURL = $_REQUEST['statsURL'];
```

```
12  + if (empty($statsURL) || $statsURL == "php://input" || !preg_match("/^http/",
      $statsURL)) {
13  +     _log('this is not a URL ');
14  +     exit;
15  + }
16  +
17  + ini_set('display_errors', 1);
18  + ini_set('display_startup_errors', 1);
19  + error_reporting(E_ALL & ~E_DEPRECATED);
20  +
21  + _log('Starting try to get URL ' . $statsURL);
22  +
23  + $result = url_get_contents($statsURL, 2);
24  +
25  + if ($result) {
26  +     _log('<span style="background-color: green; padding: 1px 4px; color:
      #FFF;">SUCCESS</span>');
27  + } else {
28  +     _log('<span style="background-color: red; padding: 1px 4px; color:
      #FFF;">FAIL</span>');
29  + }
30  +
31  + _log('Finish try to get URL ');
32  +
33  + $timeElapsed = number_format(microtime(true) - $timeStarted, 5);
34  + if ($timeElapsed>=2) {
35  +     _log('IMPORTANT: your stats took longer than 2 seconds to respond, the
      Streamer has a 2 seconds timeout rule ');
36  + }
37  +
38  + function url_get_contents($url, $timeout = 0)
39  + {
40  +     _log('url_get_contents start timeout=' . $timeout);
41  +     $agent = "AVideoStreamer";
42  +
43  +     $opts = [
44  +         'http' => ['header' => "User-Agent: {$agent}\r\n"],
45  +         "ssl" => [
46  +             "verify_peer" => false,
47  +             "verify_peer_name" => false,
```

```
48 +             "allow_self_signed" => true,
49 +         ],
50 +     ];
51 +     if (!empty($timeout)) {
52 +         ini_set('default_socket_timeout', $timeout);
53 +         $opts['http'] = ['timeout' => $timeout];
54 +     }
55 +
56 +     $context = stream_context_create($opts);
57 +
58 +     if (ini_get('allow_url_fopen')) {
59 +         try {
60 +             $tmp = file_get_contents($url, false, $context);
61 +             _log('file_get_contents:: '.htmlentities($tmp));
62 +             if (empty($tmp)) {
63 +                 _log('file_get_contents fail return an empty content');
64 +                 return false;
65 +             } else {
66 +                 _log('file_get_contents works');
67 +                 return true;
68 +             }
69 +         } catch (ErrorException $e) {
70 +             _log('file_get_contents fail catch error: ' . $e->getMessage());
71 +             return false;
72 +         }
73 +     } elseif (function_exists('curl_init')) {
74 +         _log('allow_url_fopen is NOT enabled but curl_init is, we will try
     CURL');
75 +         $ch = curl_init();
76 +         curl_setopt($ch, CURLOPT_USERAGENT, $agent);
77 +         curl_setopt($ch, CURLOPT_URL, $url);
78 +         curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
79 +         curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
80 +         curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);
81 +         if (!empty($timeout)) {
82 +             curl_setopt($ch, CURLOPT_TIMEOUT, $timeout);
83 +             curl_setopt($ch, CURLOPT_CONNECTTIMEOUT, $timeout + 10);
84 +         }
85 +         $output = curl_exec($ch);
86 +         curl_close($ch);
```

```
87  +         _log('curl_init:: '.htmlentities($output));
88  +         if (empty($output)) {
89  +             _log('curl_init fail to download');
90  +             return false;
91  +         } else {
92  +             _log('curl_init success to download');
93  +             return true;
94  +         }
95  +     } else {
96  +         _log('IMPORTANT: allow_url_fopen is NOT enabled also curl_init is NOT
      enable, please investigate it and make sure it is enabled');
97  +     }
98  +
99  +
100 +     _log('Try wget');
101 +     // try wget
102 +     $tmpDir = sys_get_temp_dir();
103 +     if (empty($tmpDir)) {
104 +         _log('IMPORTANT: your sys_get_temp_dir is empty');
105 +         return false;
106 +     }
107 +     if (!is_writable($tmpDir)) {
108 +         _log('IMPORTANT: we cannot write in your temp directory ' . $tmpDir);
109 +         return false;
110 +     }
111 +     $tmpDir = rtrim($tmpDir, DIRECTORY_SEPARATOR) . DIRECTORY_SEPARATOR;
112 +
113 +     $filename = $tmpDir . md5($url);
114 +     if (wget($url, $filename)) {
115 +         $result = file_get_contents($filename);
116 +         _log('wget:: '.htmlentities($result));
117 +         unlink($filename);
118 +         if (!empty($result)) {
119 +             _log('wget works ');
120 +             return true;
121 +         } else {
122 +             _log('wget fail ');
123 +         }
124 +     }
125 +     unlink($filename);
```

```
126 +
127 +     return false;
128 + }
129 +
130 + function wget($url, $filename)
131 + {
132 +     if (empty($url) || $url == "php://input" || !preg_match("/^http/", $url)) {
133 +         _log('this is not a URL ');
134 +         return false;
135 +     }
136 +     if (strtoupper(substr(PHP_OS, 0, 3)) === 'WIN') {
137 +         _log('this is a windows OS ');
138 +         return false;
139 +     }
140 +     $cmd = "wget --tries=1 " . escapeshellarg($url) . " -O " .
      escapeshellarg($filename) . " --no-check-certificate";
141 +     exec($cmd);
142 +     if (!file_exists($filename)) {
143 +         _log('wget download fail, we cannot read the file: ' . $filename);
144 +         return false;
145 +     }
146 +     if (empty(filesize($filename))) {
147 +         _log('wget download fail, the file is empty: ' . $filename);
148 +         return false;
149 +     } else {
150 +         _log('wget download success, the file is NOT empty: ' . $filename);
151 +         return true;
152 +     }
153 + }
154 +
155 + function _log($msg)
156 + {
157 +     global $timeStarted;
158 +     $timeElapsed = number_format(microtime(true) - $timeStarted, 5);
159 +     echo '[' . date('Y-m-d H:i:s') . "] Time Elapsed: {$timeElapsed} seconds -
      " . $msg . '<br>' . PHP_EOL;
160 + }
```

Comments 0

Please sign in to comment.