

php-cron-requests-events 🗹 (main)



Overview History Settings

Created Thu 12th Jan 2023

Snapshot taken by snyk.io a few seconds ago

| Retest now

IMPORTED BY

Е Евгений

PROJECT OWNER



Add a project owner

ANALYSIS SUMMARY

4 analyzed files (31%) Repo breakdown



Q Search...

19 of 19 issues

Group by none ✓ Sort by highest severity ✓

Command Injection

822

SCORE

SNYK CODE CWE-78 [☑]

```
163
             if(
                 is_callable("shell_exec") &&
164
                 $wget !== ''
165
             ) {
166
                 if($protocol === 'https') shell exec($wget . ' -T 1 --no-check-c
167
```

Unsanitized input from an HTTP header flows into shell_exec, where it is used to build a shell command. This may result in a Command Injection vulnerability.

Cron.php

16 steps in 1 file





822

SNYK CODE CWE-78 [☑]

```
is_callable("shell_exec") &&

swget !== ''

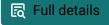
if($protocol === 'https') shell_exec($wget . ' -T 1 --no-check-celse shell_exec($wget . ' -T 1 --delete-after -q "' . $cron_url
```

Unsanitized input from an HTTP header flows into shell_exec, where it is used to build a shell command. This may result in a Command Injection vulnerability.

Cron.php

16 steps in 1 file





H Command Injection

822

SNYK CODE CWE-78 [☑]

Unsanitized input from an HTTP header flows into shell_exec, where it is used to build a shell command. This may result in a Command Injection vulnerability.

Cron.php

16 steps in 1 file





H Command Injection

822

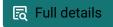
SNYK CODE CWE-78 [☑]

Unsanitized input from *an HTTP header flows* into *shell_exec*, where it is used to build a shell command. This may result in a Command Injection vulnerability.

Cron.php

16 steps in 1 file





H Cross-site Scripting (XSS)

806

SNYK CODE CWE-79 Z

Unsanitized input from *an HTTP header flows* into *the echo statement*, where it is used to render an HTML page returned to the user. This may result in a Cross-Site Scripting attack (XSS).

Cron.php 16 steps in 1 file

NEW

igorealthing Learn about this type of vulnerability and how to fix it $^{\mbox{$\sigma$}}$





SCORE 806

SNYK CODE CWE-23 [☑]

```
if($protocol === 'https') shell_exec($curl . ' -I -k --connect-1
else shell_exec($curl . ' -I --connect-timeout 1 "' . $cron_url
else {
    @fclose(
    @fopen(
```

Unsanitized input from *an HTTP header flows* into *fopen*, where it is used as a path. This may result in a Path Traversal vulnerability and allow an attacker to open arbitrary files.

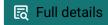
Cron.php

14 steps in 1 file



♦ Learn about this type of vulnerability and how to fix it





M Deserialization of Untrusted Data

score 603

SNYK CODE CWE-502 [☑]

Unsanitized input from *data from a remote resource flows* into *unserialize*, where it is used to deserialize an object. This may result in an Unsafe Deserialization vulnerability.

Cron.php

16 steps in 1 file





M Deserialization of Untrusted Data

SCORE

603

SNYK CODE CWE-502 [☑]

Unsanitized input from *data from a remote resource flows* into *unserialize*, where it is used to deserialize an object. This may result in an Unsafe Deserialization vulnerability.

Cron.php

16 steps in 1 file





M Deserialization of Untrusted Data

score 603

SNYK CODE CWE-502 Z

```
// examples use adressed mode

if(is_array($boot) && count($boot['handlers']) === 1): // first }

// example 1, get first element

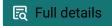
$value= unserialize(queue_address_pop($frame_size, $index_data)
```

Unsanitized input from *data from a remote resource flows* into *unserialize*, where it is used to deserialize an object. This may result in an Unsafe Deserialization vulnerability.

Cron.php

16 steps in 1 file





M Deserialization of Untrusted Data

score 603

SNYK CODE CWE-502 [3]

```
//usleep(2000); // test load, micro delay
//usleep(2000); // test load, micro delay
// sample 2, get last - 10 element, and get first frame in (
yvalue= unserialize(queue_address_pop($frame_size, $index_data)
```

Unsanitized input from *data from a remote resource flows* into *unserialize*, where it is used to deserialize an object. This may result in an Unsafe Deserialization vulnerability.

Cron.php

16 steps in 1 file





M Deserialization of Untrusted Data

603

SNYK CODE CWE-502 2

Unsanitized input from *data from a remote resource flows* into *unserialize*, where it is used to deserialize an object. This may result in an Unsafe Deserialization vulnerability.

Cron.php

16 steps in 1 file





M Deserialization of Untrusted Data

score 603

SNYK CODE CWE-502 Z

Unsanitized input from *data from a remote resource flows* into *unserialize*, where it is used to deserialize an object. This may result in an Unsafe Deserialization vulnerability.

🕠 cron.php

16 steps in 1 file





M Deserialization of Untrusted Data

603

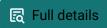
SNYK CODE CWE-502 [☑]

Unsanitized input from *data from a remote resource flows* into *unserialize*, where it is used to deserialize an object. This may result in an Unsafe Deserialization vulnerability.

Cron.php

16 steps in 1 file





M Deserialization of Untrusted Data

SCORE

603

SNYK CODE CWE-502 Z

```
// execution time: 0.051764011383057 end - start, 1000 cycles

while(true){ // example: loop from the end

frame= queue_address_pop($frame_size, PHP_INT_MAX, '', "coustable start);

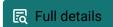
yvalue= unserialize($frame);
```

Unsanitized input from *data from a remote resource flows* into *unserialize*, where it is used to deserialize an object. This may result in an Unsafe Deserialization vulnerability.

Cron.php

17 steps in 1 file





M Deserialization of Untrusted Data

score 603

SNYK CODE CWE-502 [☑]

Unsanitized input from *data from a remote resource flows* into *unserialize*, where it is used to deserialize an object. This may result in an Unsafe Deserialization vulnerability.

Cron.php

8 steps in 1 file





SCORE 603

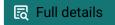
SNYK CODE CWE-502 [☑]

Unsanitized input from data from a remote resource flows into unserialize, where it is used to deserialize an object. This may result in an Unsafe Deserialization vulnerability.

Cron.php

9 steps in 1 file





M Use of Password Hash With Insufficient Computational Effort 566

SNYK CODE CWE-916 [□]

```
if(isset($init[$job_process_id])) return;
finit[$job_process_id]= true;
finit[$job_process_id]= true;
finit[$job_process_id]['md5'])) {
    if(isset($cron_session[$job_process_id]['md5'])) {
        if($cron_session[$job_process_id]['md5'] !== md5(serialize($job));
}
```

MD5 hash (used in md5) is insecure. Consider changing it to a secure hashing algorithm.

Cron.php

1 step in 1 file





M Use of Password Hash With Insufficient Computational Effort 566

if(isset(\$cron_session[\$job_process_id]['md5'])) {
 if(\$cron_session[\$job_process_id]['md5'] !== md5(serialize(\$job));
}
cron_session[\$job_process_id]['md5'] = md5(serialize(\$job));

MD5 hash (used in md5) is insecure. Consider changing it to a secure hashing algorithm.



1 step in 1 file





M Use of Password Hash With Insufficient Computational Effort 566

SNYK CODE CWE-916 [☑]

```
$\text{$\scron_session[$job_process_id]['last_update']= 0;}$$
$\text{$\scron_session[$job_process_id]['complete']= false;}$$
$\text{$\gamma_02 \\ }$$
$\text{$\scron_session[$job_process_id]['md5']= md5(serialize($job));}$$
$\text{$\scron_session[$job_process_id]['md5']= md5(serialize($job));}$$
$\text{$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_session[$\scron_se
```

MD5 hash (used in md5) is insecure. Consider changing it to a secure hashing algorithm.

Cron.php

1 step in 1 file



