Concrete5 8.5.5 Phar Deserialization

Authored by EgiX | Site karmainsecurity.com

Posted Jul 20, 2021

Concrete5 versions 8.5.5 suffer from a logging settings phar deserialization vulnerability. User input passed through the logFile request parameter is not properly sanitized before being used in a call to the file_exists() function at line 91. This can be exploited by malicious users to inject arbitrary PHP objects into the application scope (PHP Object Injection via phar:// stream wrapper), allowing them to carry out a variety of attacks, such as executing arbitrary PHP code. Successful exploitation of this vulnerability requires an administrator account.

tags | advisory, arbitrary, php
advisories | CVE-2021-36766
SHA-256 | 4737c6d7d22010e52296503aaa366abc55f04d975b7b1fd092c8c80e1a164e8a

Change Mirror Download

```
------------------------------------------------------------------------
Concrete5 <= 8.5.5 (Logging Settings) Phar Deserialization Vulnerability
------------------------------------------------------------------------


[-] Software Link:

https://www.concrete5.org


[-] Affected Versions:

Version 8.5.5 and prior versions.


[-] Vulnerability Description:

The vulnerable code is located within the
/concrete/controllers/single_page/dashboard/system/environment/logging.php
script. Specifically, into the Logging::update_logging()  method:

61.      public function update_logging()
62.      {
63.          $config = $this->app->make('config');
64.          $request = $this->request;
65.
66.          if (!$this->token->validate('update_logging')) {
67.              return
$this->showError($this->token->getErrorMessage());
68.          }
69.
70.          // Load in variables from the request
71.          $mode = (string) $request->request->get('logging_mode') ===
'advanced' ? 'advanced' : 'simple';
72.          $handler = $mode === 'simple' ? (string)
$request->request->get('handler', 'database') : null;
73.          $logFile = $handler === 'file' ? (string)
$request->request->get('logFile') : null;
74.          $enableDashboardReport =
$request->request->get('enable_dashboard_report') ? true : false;
75.          $loggingLevel = strtoupper((string)
$request->request->get('logging_level'));
76.          $intLogErrorsPost =
$request->request->get('ENABLE_LOG_ERRORS') === 1 ? 1 : 0;
77.          $intLogEmailsPost =
$request->request->get('ENABLE_LOG_EMAILS') === 1 ? 1 : 0;
78.          $intLogApiPost = $request->request->get('ENABLE_LOG_API')
=== 1 ? 1 : 0;
79.
80.
81.          // Handle 'file' based logging
82.          if ($handler === 'file') {
83.              $directory = dirname($logFile);
84.
85.              // Validate the file name
86.              if (pathinfo($logFile, PATHINFO_EXTENSION) !== 'log') {
87.                  return $this->showError(t('The filename provided
must be a valid filename and end with .log'));
88.              }
89.
90.              // Validate the file path, create the log file if needed
91.              if (!file_exists($logFile)) {


User input passed through the "logFile" request parameter is not
properly sanitized before being used in a call to the file_exists()
function at line 91. This can be exploited by malicious users to inject
arbitrary PHP objects into the application scope (PHP Object Injection
via phar:// stream wrapper), allowing them to carry out a variety of
attacks, such as executing arbitrary PHP code. Successful exploitation
of this vulnerability requires an administrator account.


[-] Solution:

No official solution is currently available.


[-] Disclosure Timeline:

[20/12/2020] - Vendor notified through HackerOne
[22/12/2020] - Vendor asks suggestions to fix the issue, feedback
provided
[18/03/2021] - Version 8.5.5 released, vulnerability not fixed
[02/06/2021] - Asked for an update, no response
[06/07/2021] - Asked for an update, no response
[16/07/2021] - CVE number assigned
[19/07/2021] - Public disclosure


[-] CVE Reference:

The Common Vulnerabilities and Exposures project (cve.mitre.org)
has assigned the name CVE-2021-36766 to this vulnerability.


[-] Credits:

Vulnerability discovered by Egidio Romano.


[-] Other References:

https://hackerone.com/reports/1063039


[-] Original Advisory:

http://karmainsecurity.com/KIS-2021-05
```

Spoof (2,166)
SQL Injection (16,102)
TCP (2,379)
Trojan (686)
UDP (876)
Virus (662)
Vulnerability (31,136)
Web (9,365)
Whitepaper (3,729)
x86 (946)
XSS (17,494)
Other

SUSE (1,444)
Ubuntu (8,199)
UNIX (9,159)
UnixWare (185)
Windows (6,511)
Other

Site Links

News by Month
News Tags
Files by Month
File Tags
File Directory

About Us

History & Purpose
Contact Information
Terms of Service
Privacy Statement
Copyright Information

Hosting By

Rokasec

Follow us on Twitter

Subscribe to an RSS Feed