All You Need to Know About CSV Injection and Excel Macro Injection Attacks

By Sandeep Kamble

Published: 2021-01-10 · Archived: 2026-04-05 19:48:52 UTC

What is CSV Injection?

CSV Excel Macro Injection, also known as Formula Injection or CSV Injection, is an attack technique that we use in the day-to-day penetration testing of the application.

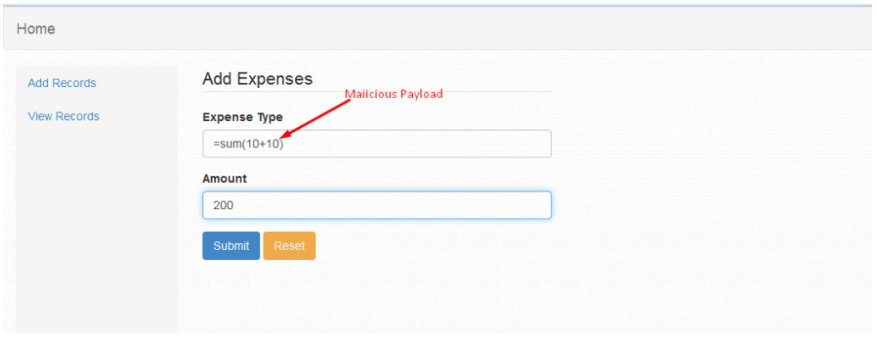
CSV injection is a vulnerability that affects applications that have the export spreadsheet functionality. These spreadsheets generate dynamically from invalidated or unfiltered user inputs. Modern web applications offer spreadsheet export functions these days. This allows the user to download data in a .csv file format or .xls file format. This is suitable for handling spreadsheet applications like MS-Excel and OpenOffice Calc, as a result of which the cells in the spreadsheets can contain inputs from untrusted sources. As a result, the end-user who is accessing the exported spreadsheet can be affected.
This vulnerability can be used by an attacker to execute attacks such as client-side command injection or code injection. Basically, the attack scenario for this is purely targeting the user(s) who download the Excel file naturally. We usually disregard this attack as a non-issue. However, websites should still be aware that the information they are exporting can potentially affect the users.

Demonstration of CSV excel macro injection:

CSV excel macro injection can be exploited when the application supports export to excel functionality.
This happens in spreadsheet files, which dynamically generate from invalid input data.

CSV Injection Payloads used to test and exploit:

We can use formulas, which we use in excel for carrying out operations to test formula injection on websites.
Eg: =sum(10+10)



Example of CSV Injection:

As you can see, once we click on the export excel option, the records automatically export to an excel file with the .xls format. Thus, this allows us to download the .xls file

Technical Analysis of the Vulnerability:

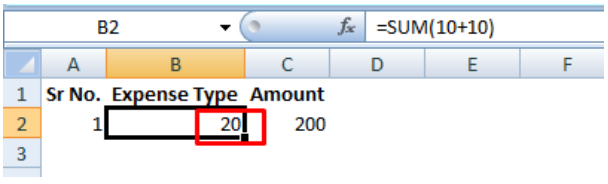
This vulnerability occurs due to the concept of dynamic data exchange (DDE). DDE is a protocol for interprocess communication under Windows supported by MS-Excel, LibreOffice, and Apache OpenOffice.

DDE Function Format:

The DDE function is in the following format:

=DDE(server; file; item; mode)

So by using some malicious arguments, it is possible to remotely execute applications or commands on the victim's computer of whoever opens the document.



Common CSV Injection Payload

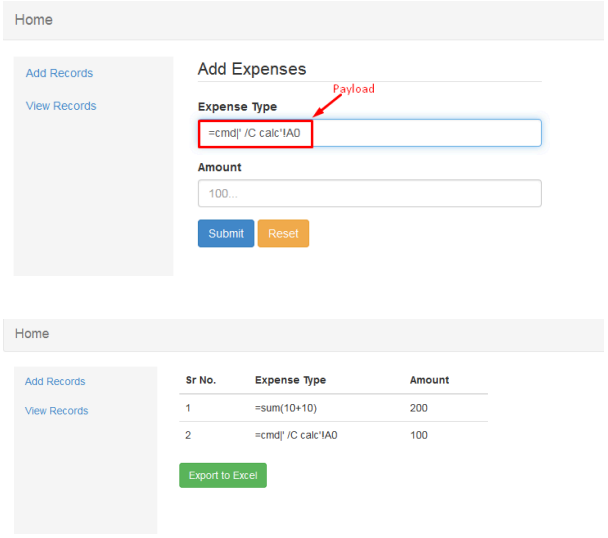
So, the most common CSV Injection payload used is:
=cmd|' /C calc'!A0

Technical Details of the above CSV Injection payload:

- cmd: The name the server can respond to whenever a client is trying to access the server.
- /C calc: The specific file or command name—in this case, 'calc' (i.e., calc.exe).
- !A0: The item name that specifies the unit of data that a server can respond to when the client is requesting the data.

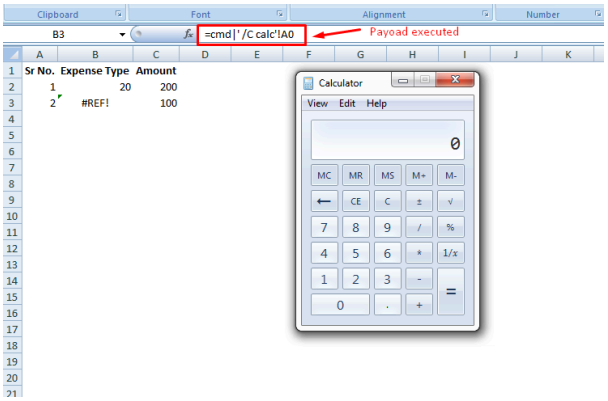
So our final DDE call becomes:

DDE ("cmd";"/C calc";"!A0")



As you can see in the above screenshots, our payload adds to the input fields. Now, once we export this record to an excel file, our payload informs the program to run an application called cmd.exe with the command /C calc which executes calc.exe from the command line.

Once the excel file is open calc.exe will be executed as shown in the screenshot below.



Alternative Payloads:

Mostly, these payloads are all fine and well. But, sometimes the `=` character filters out. However, we can use some different combinations such as @, + or -. So, the current payload of choice for exploiting this as a proof of concept is:

```
@SUM(1+1)*cmd|' /C calc'!A0
```

We can use any formula starting with:

- =
- +
- –
- @

Recommendations

In conclusion, I recommend that it is always a good practice not to trust user inputs and to always encode the output. Also, for the successful execution of the formula, an attacker will have to use the '-', '=', and the pipe (|) is used to execute the binary in the excel software. Hence, it is strongly recommended to filter the '- ', '|', '+', and '=' to mitigate this vulnerability.

References

[1] Download Vulnerable Code here

[2] CSV Injection : https://blog.zsec.uk/csv-dangers-mitigations/

[3] Comma Separated Vulnerabilities
