Satyam Singh Follow
Jul 14, 2021 · 6 min read · Listen

Vulnerabilities in Booking Core 1.7

Hello everyone and welcome to my first blog on Medium. I have been writing blogs on the information security domain which can be found at resources.infosecinstitute.com and pentest-tools.com.

In this blog, I am going to cover few important vulnerabilities on the Booking Core application that I discovered while doing the bug bounty for one of the event management and booking applications. The vulnerabilities mentioned in this blog are affecting the booking core version 1.7.

Booking Core is a Booking System based on Laravel, designed for a travel website, Marketplace, Travel Agency, Tour Operator, Room Bnb, Villa Rental, Resort Rental, Make Travel website.

Let's discuss various issues found and reported on this application. Each vulnerability in this blog has been assigned with CVE ID and details of the same are mentioned in each section.

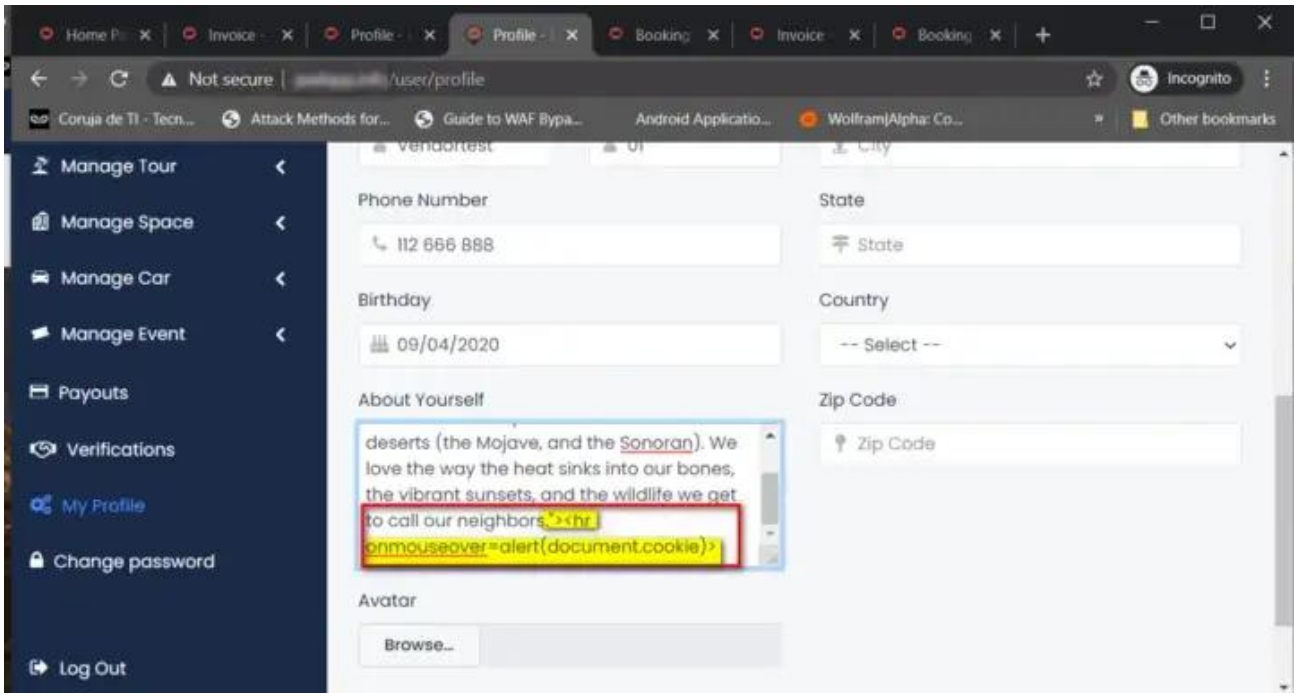
Issue 1: Cross-Site Scripting Attack

Well, we all know what is cross-site scripting aka XSS. This vulnerability allows an adversary to inject and run JavaScript on the victim browser, resulting in the execution of the malicious script, installing Trojans, stealing the session token of a logged-in user, redirecting to another website, or anything which a JavaScript is capable of doing.

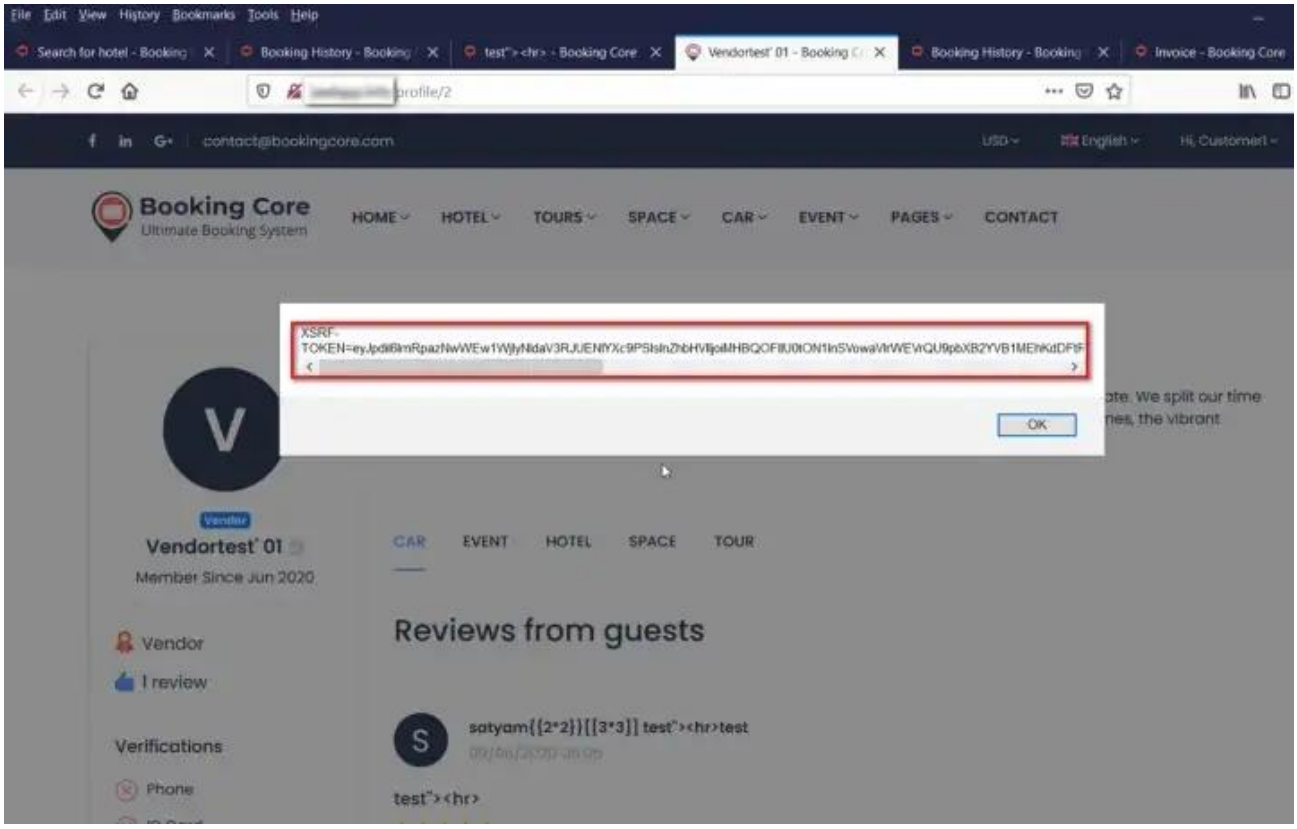
The XSS issues were identified on various fields of multiple pages and a request was raised with https://cve.mitre.org/ for CVE ID assignment.

CVE Details

> ***CVE ID Assigned:*** *CVE-2020–25444*
>
> ***Product & Version:*** *Booking Core 1.7*
>
> ***Problem Type:*** *The application is vulnerable to Cross-Site Scripting attacks.*
>
> ***Description:*** *The Booking Core application was found to be vulnerable to XSS attacks. The "About Yourself" section under the "My Profile" page is vulnerable to Cross-Site Scripting (XSS) attacks. Besides, the following pages and fields were found to be affected by the same issue:*
>
> *"Hotel Policy" field under the "Hotel Details" page*
>
> *"Pricing code" and "name" fields under the "Manage Tour" page*
>
> *All the labels under the "Menu" section*

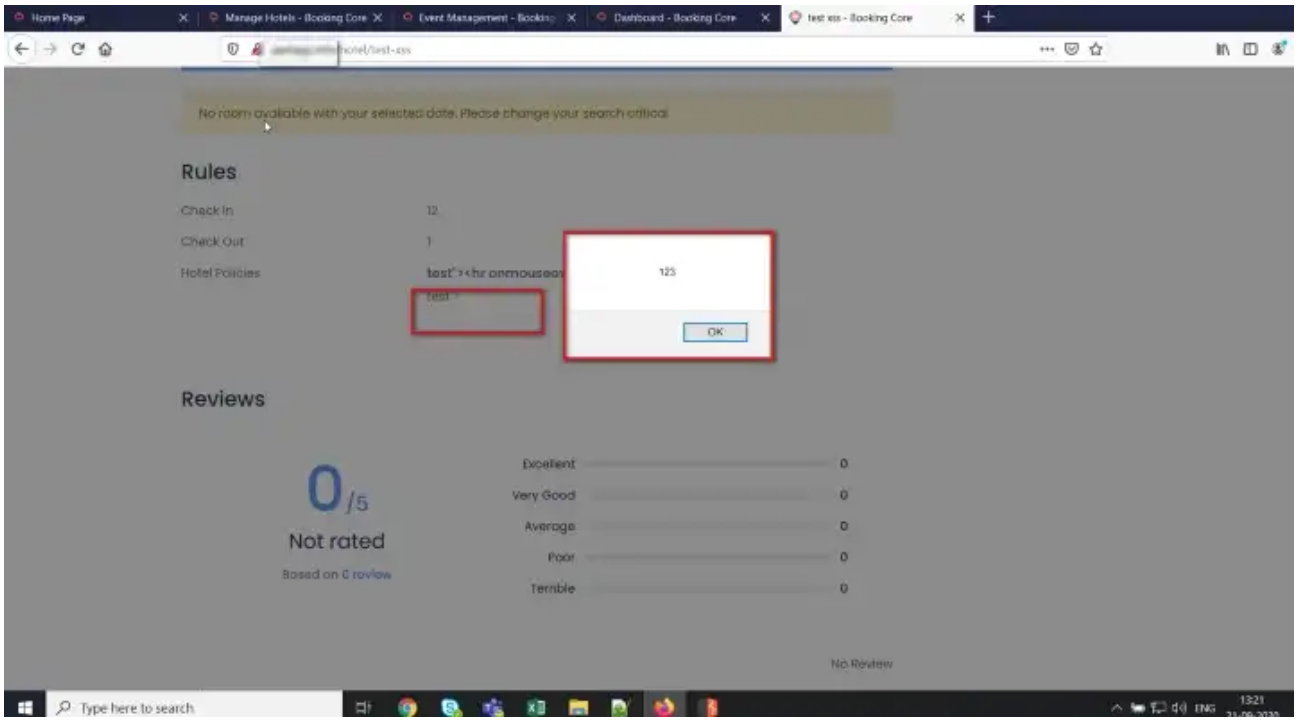


XSS payload in "About Yourself" field under user profile page

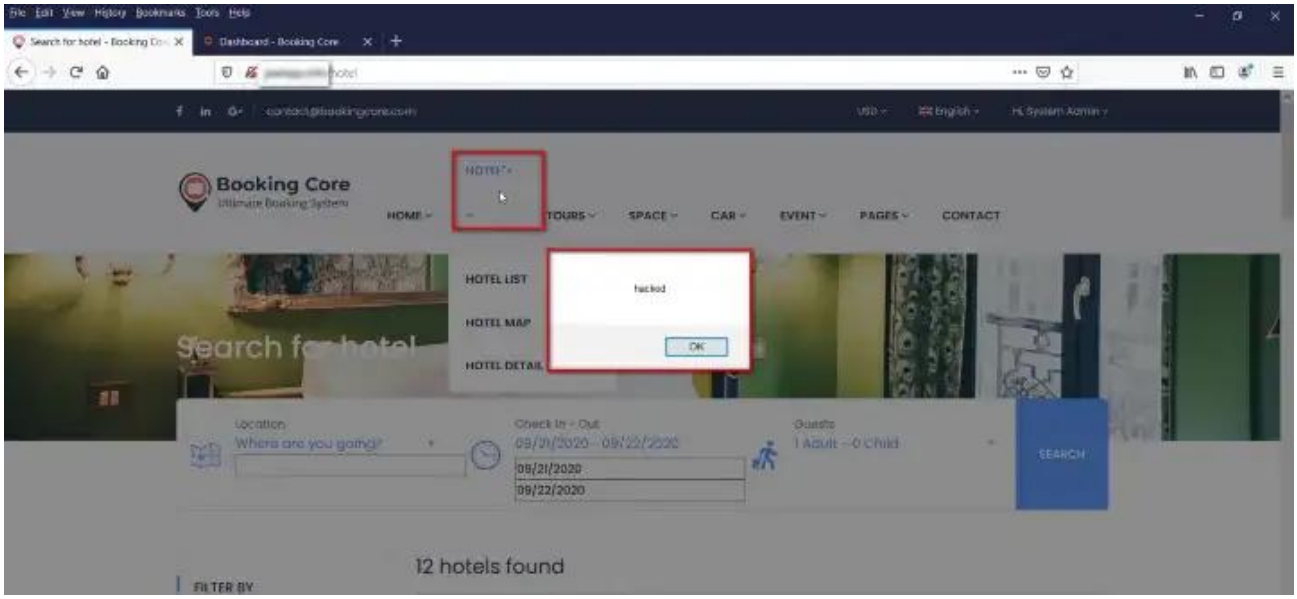


XSS payload execution

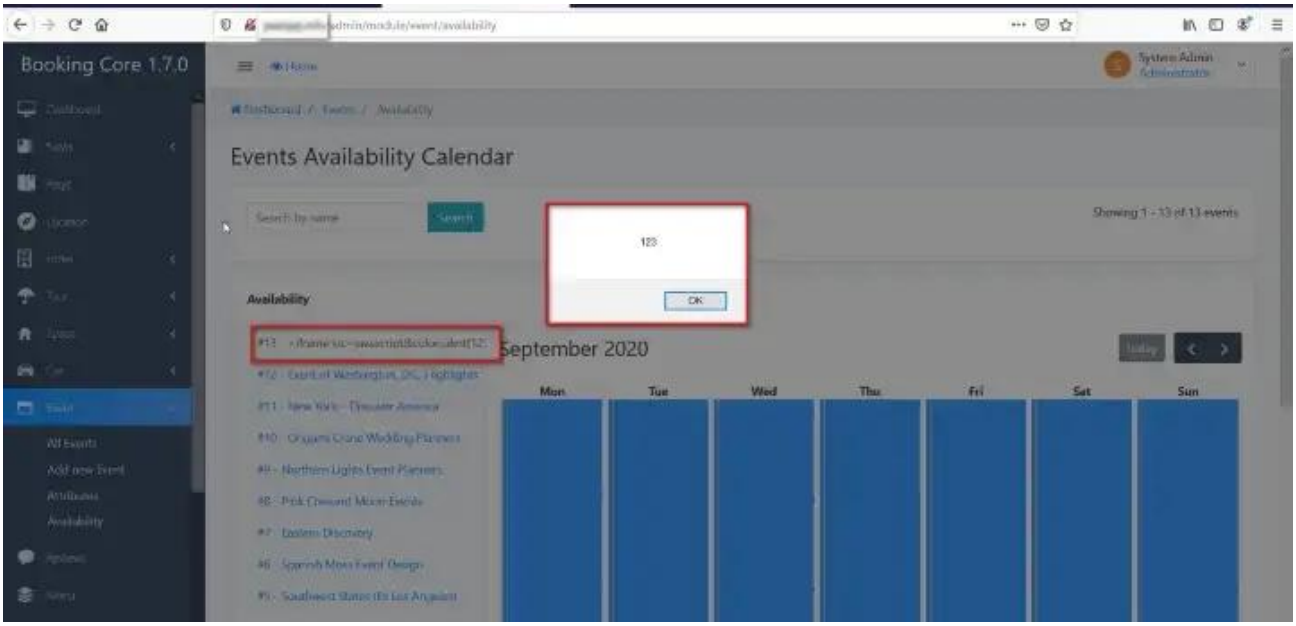
Similarly, other pages mentioned in the description section are vulnerable to XSS attacks.



XSS payload execution on the "Hotel Policy" field



XSS payload execution on label names



XSS payload execution on “Pricing code” field

Issue 2: Account takeover using Cross-Site Request Forgery attack

Cross-Site Request Forgery aka CSRF is again a high-risk issue that can lead to account takeover. This is how the attack works: an attacker creates a special page/URL and tricks the user into visiting it while the user is logged in to the application. This special page triggers a request to the application with the user’s session information. This request may be for changing the email ID.

The request is forged to look like a valid request for this operation. All the details required for the operation to succeed are present as query-string/POST variables. When the request is sent from the victim’s machine, valid cookies with the session information are also sent.

The application misunderstands that this request is valid, as it contains the cookies. So, the operation succeeds without the user’s knowledge.

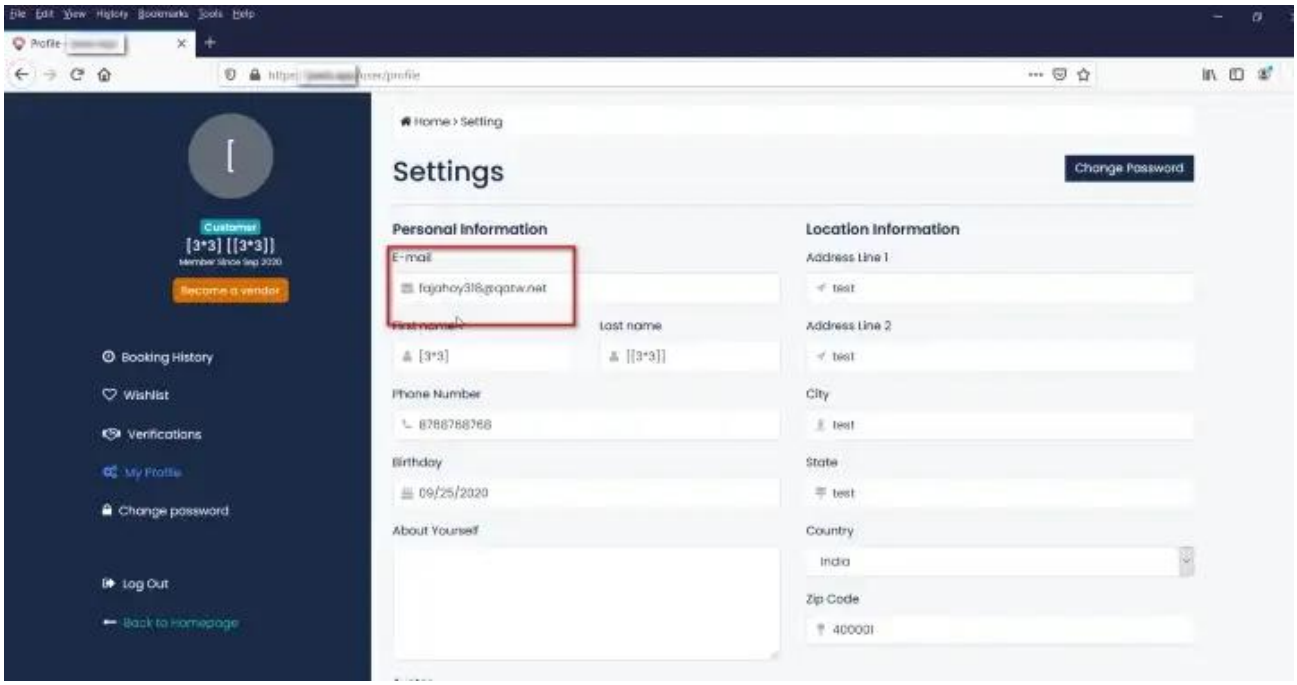
The special page is quite easy to create. It might be a simple HTML page with an < img alt=” “ src=” “ /> tag with the source pointing to the page that operates.

Note 1: The Booking Core application was found to be using CSRF token on all the POST requests but this can be bypassed by sending the POST request as a GET request. :)

Note 2: This issue applies to all the pages of the application where some modifications are being made. For example: Approving a user, change the password of a user, publish/hide a hotel, modify the content of an ad, etc.

CVE Details

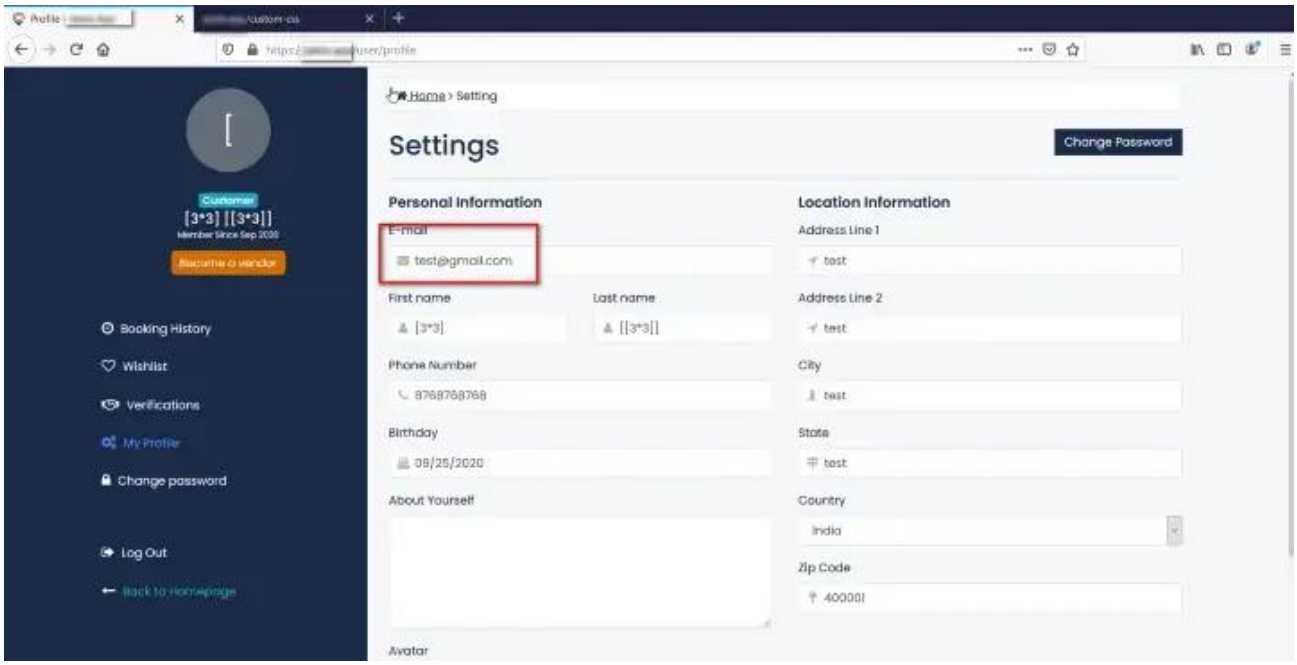
> ***CVE ID Assigned:*** *CVE-2020–27379*
>
> ***Product & Version:*** *Booking Core 1.7*
>
> ***Problem Type:*** *The application is vulnerable to Cross-Site Request Forgery (CSRF) attacks.*
>
> ***Description:*** *An adversary can take over any account using CSRF vulnerability. The CSRF token is not being validated when the request is sent as a GET method. This results in an unauthorized change in the user's email ID, which can later be used to reset the password. The new password will be sent to a modified email ID.*



The current email of the victim is “fajahoy318@aqtw.net”

Send a link to the victim with the email ID set to “test@gmail.com”

The backend request going to the server contains the attacker-supplied email ID



The email ID of the victim is changed to “test@gmail.com”

Now the email ID is updated and an attacker can request a password reset. The link to reset the password will be sent to “test@gmail.com”. This would result in an account takeover.

Issue 3: CSV Formula Injection

Many web applications allow the user to download content such as templates for invoices or user settings to a .CSV file. Many users choose to open the CSV file in either Excel, Libre Office, or Open Office. When a web application does not validate the contents of the CSV file, it could lead to the contents of a cell or many cells being executed.

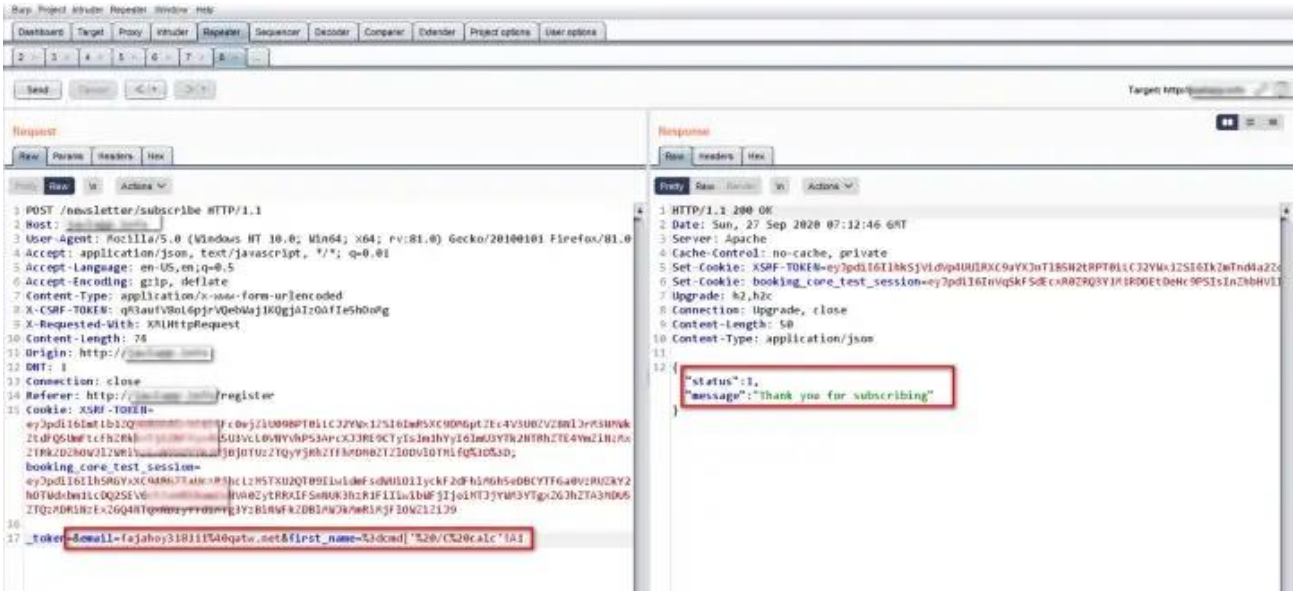
This attack exploits the trust of the user in two ways:

1. The user trusts the site that the content is coming from.

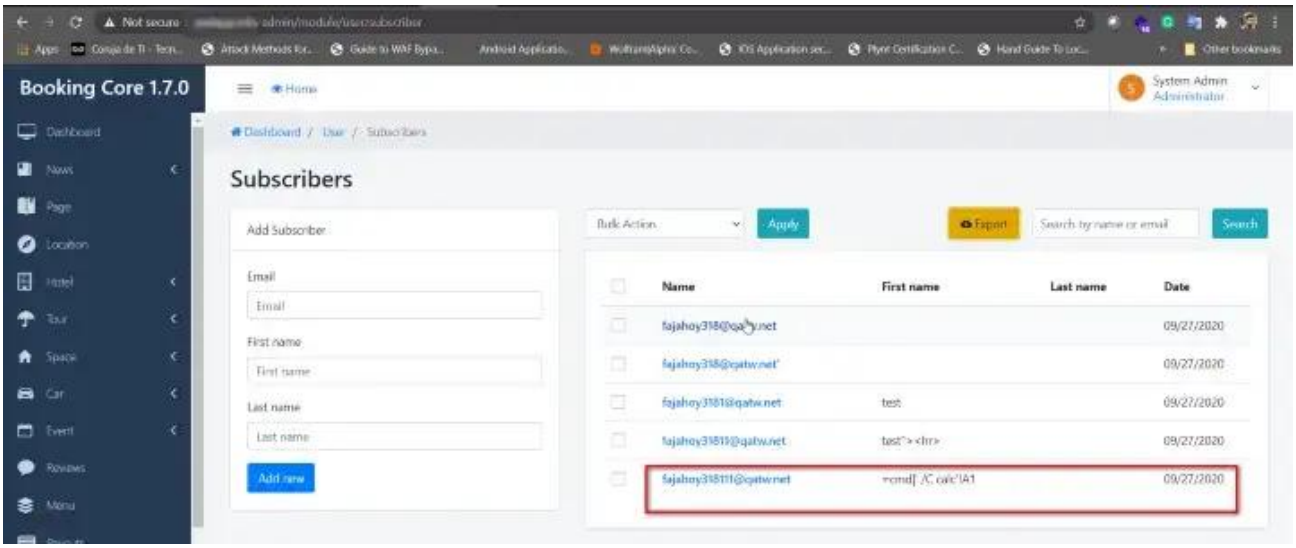
2. The user assumes that it is only a .CSV file and that it will not contain functions or macros and will not care about any warnings from Excel about a potential malicious functionality in the file.

CVE Details

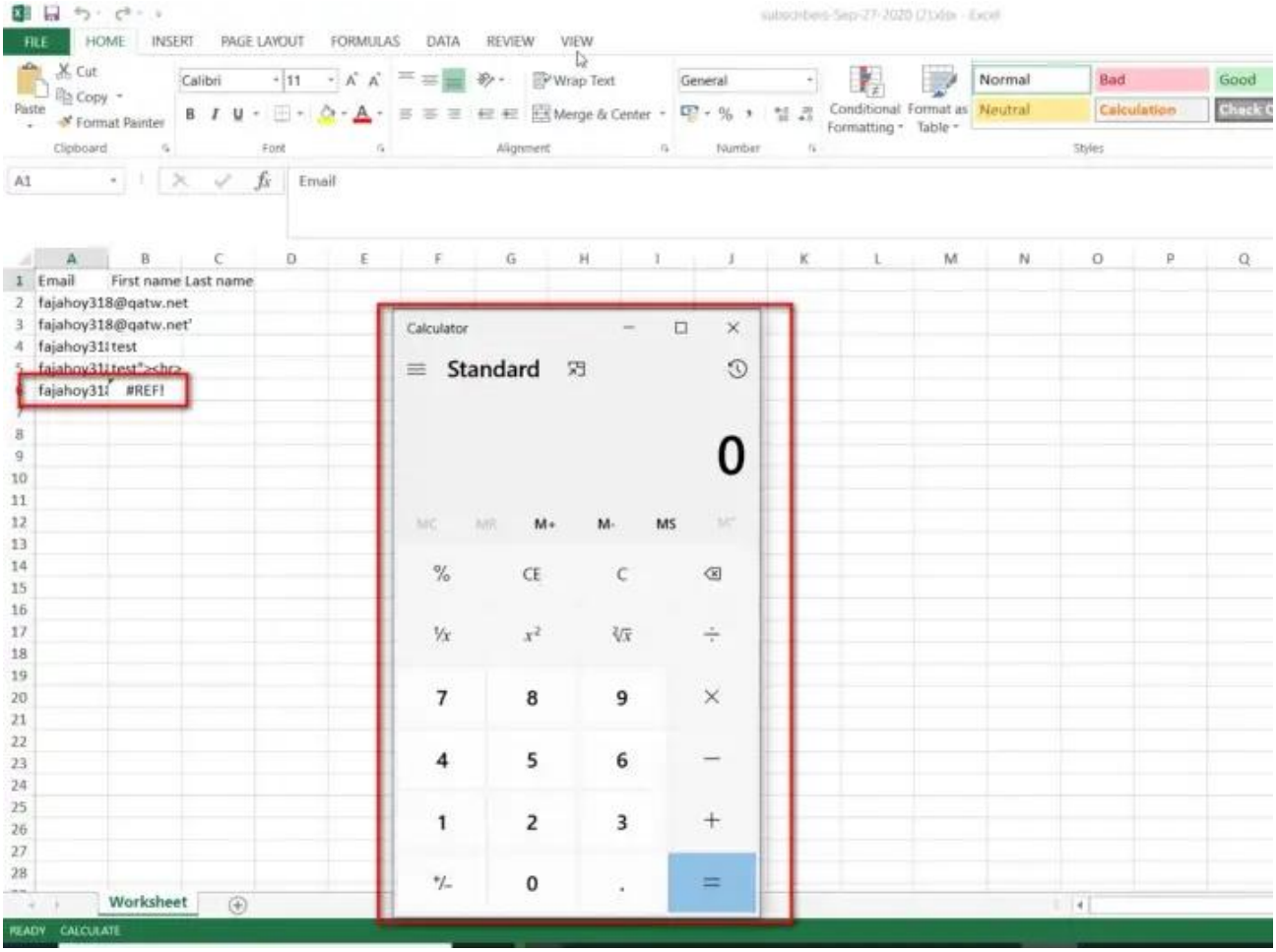
> ***CVE ID Assigned:*** *CVE-2020–25445*
>
> ***Product & Version:*** *Booking Core 1.7*
>
> ***Problem Type:*** *The application is vulnerable to CSV Formula injection attacks.*
>
> ***Description:*** *The “Subscribe” feature of the application is vulnerable to CSV formula injection. The input containing the excel formula is not being sanitized by the application. As a result, when an admin in the backend downloads and opens the CSV, the content of the cells is executed. Vulnerable fields: First name and Last name of the “Subscribe” request.*



Intercepted Burp request showing excel formula in the “first_name” field of the subscription request



Admin users can see subscription requests and have an option to export



Excel formula is executed when the exported file is opened on the local system of the Admin user

This concludes the main issues identified in this application. All the issues were reported to the product team and were fixed on the next release. The reported issues can be found in the changelog of Version 1.8.0 released on Oct 17th, 2020.

I hope you enjoyed reading this blog. I will be back with more interesting blogs in the future.

Bug Bounty Penetration Testing Infosec Web Application Security Hacking