Example 2: Reporting CSRF

`Title`: Cross-Site Request Forgery (CSRF) in Consumer Registration

`CWE`: CWE-352: Cross-Site Request Forgery (CSRF)

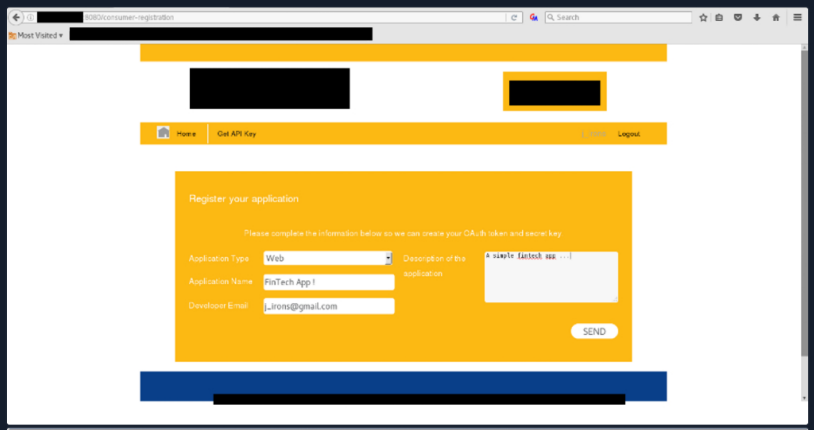
`CVSS 3.1 Score`: 5.4 (Medium)

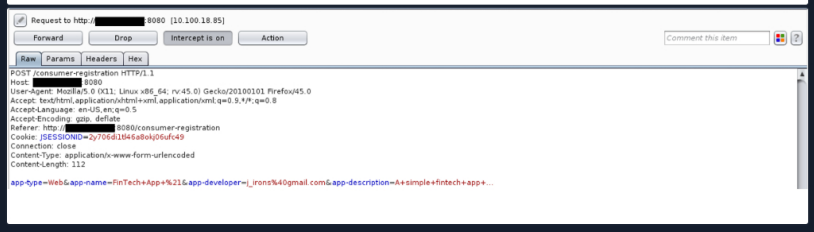
`Description`: During our testing activities, we identified that the web page responsible for consumer registration is vulnerable to Cross-Site Request Forgery (CSRF) attacks. Cross-Site Request Forgery (CSRF) is an attack where an attacker tricks the victim into loading a page that contains a malicious request. It is malicious in the sense that it inherits the identity and privileges of the victim to perform an undesired function on the victim's behalf, like change the victim's e-mail address, home address, or password, or purchase something. CSRF attacks generally target functions that cause a state change on the server but can also be used to access sensitive data.

`Impact`: The impact of a CSRF flaw varies depending on the nature of the vulnerable functionality. An attacker could effectively perform any operations as the victim. Because the attacker has the victim's identity, the scope of CSRF is limited only by the victim's privileges. Specifically, an attacker can register a fintech application and create an API key as the victim in this case.

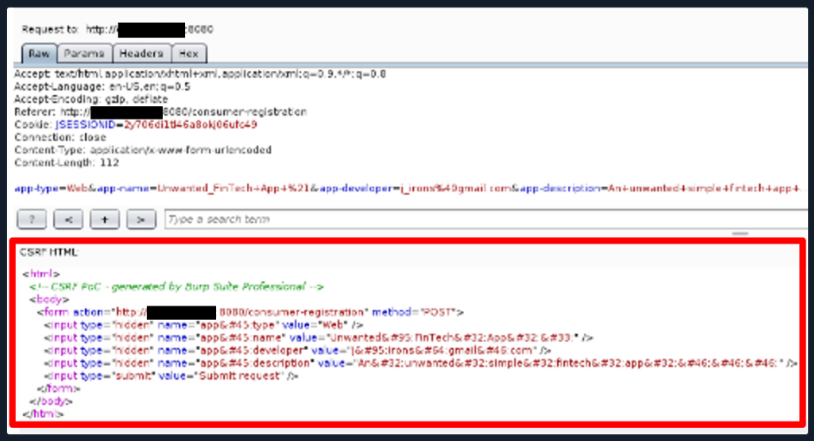
`POC`:

Step 1: Using an intercepting proxy, we looked into the request to create a new fintech application. We noticed no anti-CSRF protections being in place.

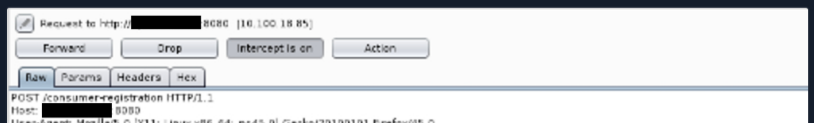




Step 2: We used the abovementioned request to craft a malicious HTML page that, if visited by a victim with an active session, a cross-site request will be performed, resulting in the advertent creation of an attacker-specific fintech application.



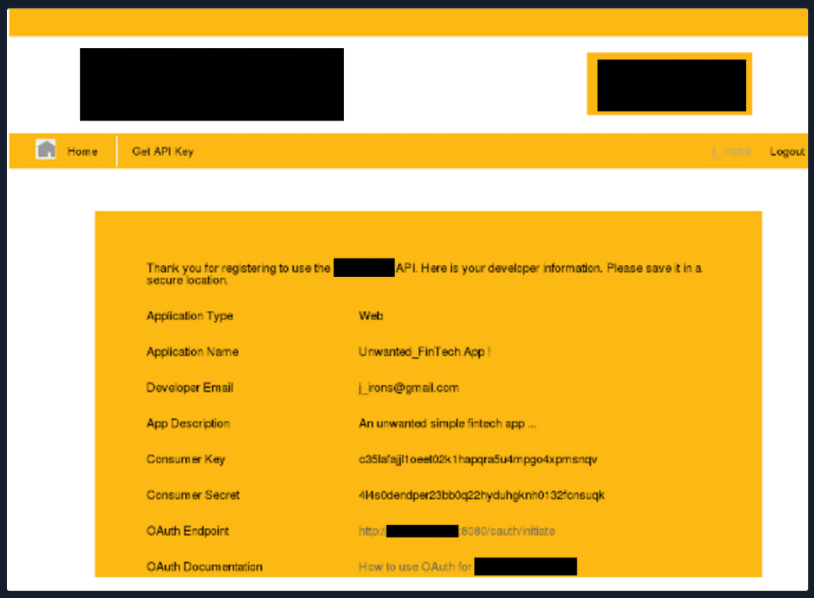
Step 3: To complete the attack, we would have to send our malicious web page to a victim having an open session. The following image displays the actual cross-site request that would be issued if the victim visited our malicious web page.



```
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://burp/show/1
Cookie: JSESSIONID=2y706dd1df46a8obj06ufc49
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 151

app-type=Web&app-name=Unwanted_FinTech+App+%21&app-developer=j_irons%40gmail.com&app-description=An+unwanted+simple+fintech+app+
```

Step 4: The result would be the inadvertent creation of a new fintech application by the victim. It should be noted that this attack could have taken place in the background if combined with finding 6.1.1. <-- 6.1.1 was an XSS vulnerability.



CVSS Score Breakdown

Attack Vector:	Network - The attack can be mounted over the internet.
Attack Complexity:	Low - All the attacker has to do is trick a user that has an open session into visiting a malicious website.
Privileges Required:	None - The attacker needs no privileges to mount the attack.
User Interaction:	Required - The victim must click a crafted link provided by the attacker.
Scope:	Unchanged - Since the vulnerable component is the webserver and the impacted component is again the webserver.
Confidentiality:	Low - The attacker can create a fintech application and obtain limited information.
Integrity:	Low - The attacker can modify data (create an application) but limitedly and without seriously affecting the vulnerable component's integrity.
Availability:	None - The attacker cannot perform a denial-of-service through this CSRF attack.