

## Where I started

I was sitting in the library with my friend and we wanted to order food from "Commons Marketplace". I didn't have enough JumboCase on my account, so I went to the JumboCash page to add funds. Upon logging into the website and browsing to the payment page, I looked at the URL and saw smth like this:

```
🔒 jumbocash.net/paymentcc.php?tab=credit&skey=ab9d31754e50353770fbaab3bd3a840d&cid=233&
```

Curious, I decided to head on over to where SPII is stored, in the statements. There I saw a URL similar to this:

```
🔒 jumbocash.net/statementdetail.php?cid=233&skey=ab9d31754e50353770fbaab3bd3a840d&startdate=2023-05-01&enddate=2023-11-31&acct=2
```

From the URL you can pinpoint exactly what the *skey* is:

**ab9d31754e50353770fbaab3bd3a840d&** I thought "Surely this isn't a static key that is used for every account upon login, right?" I copied my entire URL and pasted into an incognito page, BOOM, everything was there. Palms sweating, I asked my friend to send me their URL upon logging onto JumboCash and this was her *skey*:

**7de4c8099431f2b3d79e60cb8c0602c1&.** Scared, I replaced my *skey* with hers and got this:

The screenshot shows a web browser window with the URL `jumbocash.net/index.php?skey=7de4c8099431f2b3d79e60cb8c0602c1&cid=233&`. The page header features the Tufts University logo and the JumboCash logo. Below the header is a large photograph of a Tufts University sign. The left sidebar contains navigation links: LOGIN HERE, ADD JUMBOCASH, PERSONALIZE (with sub-options: Grant Additional Access, Request Money, Low Balance Warning, Automatic Deposits, Change Student/Faculty/Staff Password), FOR BUSINESS PARTNERS (with sub-option: Sign up to have your business accept JumboCash), and NAVIGATE. The main content area is titled "JumboCash Office" and includes a sub-headline "Sign up for online access to your JumboCash account." It lists several benefits: Check your Account Balance & Activity, Make JumboCash Deposits, Find out about Updates & Specials, and See where you can use your JumboCash. A descriptive paragraph explains that JumboCash is a cashless way to pay for campus purchases like snacks, printing, copying, and laundry. It also mentions its use at local restaurants, CVS stores, and CVS in Davis Square.

"Dang it". I tried the same for the statement page "statement.php" and got a **NOI** blank page. After trying some basic SQL injection commands such as '*OR '1'='1* to no avail, I remembered using sqlmap during our CTF. I raced to our CTF writeup and found the section on SQL Injection using sqlmap. I ran sqlmap and after a few minutes of hitting "yes" to testing a certain .php page, I tried accessing JumboCash and got hit with this:

---

Your IP was locked due to too many request. Please try again after 5 mins

Then, I looked back at my terminal and saw that after each test, SQLMap sent me this warning:

```
[12:37:12] [ERROR] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--risk' options if you wish to
perform more tests. If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could try to use option '--
tamper' (e.g. '--tamper=space2comment') and/or switch '--random-agent', skipping to the next target
[12:37:12] [WARNING] HTTP error codes detected during run:
429 (Too Many Requests) - 147 times
```

Soooo, everything I had been running for the past few min was probably all useless because my IP had been blocked...

## Wrap Up Thoughts

So, after a rollercoaster of emotions, I came to this conclusion: Good job Tufts Security Team. JumboCash was protected against SQLmap scripts as it times out IPs that send too many requests to the server. As for the *skey* situation, playing around with it further led me to these observations:

- The *skey* is generated upon starting the user's session (logging into the JumboCash website)
- The *skey* is somehow protected/authenticated by the browser's cookie (Can still "login" to JumboCash page on the same browser in incognito mode, but not on a different browser or computer)

I was shocked to see how every php page uses the *skey* to validate the user's permission to view "x" page and how everything was just given/visible in the URL bar. I was especially shocked at how the website was coded as modifying the "Start/End" Dates on the statement page actually changes what is shown. Attempting to sql inject the date results in an "invalid date" message. The *cid=233&cid* is the same for everyone and modifying that results in an error page.

## Future Steps

So, although the JumboCash website is coded with "exposed" security information, it has measures behind it to protect the user and whatever SII/PII is stored in its databases. However, even though it has these security measures, having exposed security keys isn't a good practice and who knows there is some way to exploit the communication between the cookie and the skey authentication. This is what I found upon looking at the Cookies during my active session:

	Name	Value	Domain	Path	Expires...	Size	HttpOnly	Secure	SameSite	Partition...	Priority
workers	__Secure-3PSIDCC	ACA-OxM3u2LZv5ePlz7as4QRqHDEhmCY_Wp4C4j...	.google...	/	2024-1...	92	✓	✓	None		High
	__Secure-3PAPISID	FlymXbzLOrYlH9O/ANzcM2xLP4wIJZZm8	.google...	/	2024-1...	51	✓	✓	None		High
	__Secure-1PSIDTS	sids-CjIBLFra0ggcoI9TfZ64g5UwnFq0KhCHSwpq...	.google...	/	2024-0...	94	✓	✓	None		High
	__Secure-3PSIDTS	sids-CjIBLFra0ggcoI9TfZ64g5UwnFq0KhCHSwpq...	.google...	/	2024-0...	94	✓	✓	None		High
age storage	__Secure-1PAPISID	FlymXbzLOrYlH9O/ANzcM2xLP4wIJZZm8	.google...	/	2024-1...	51	✓	✓	None		High
3	SAPISID	FlymXbzLOrYlH9O/ANzcM2xLP4wIJZZm8	.google...	/	2024-1...	41	✓	✓	None		High
	SSID	ANcoZOIZP2Jg83yC	.google...	/	2024-1...	21	✓	✓	None		High
	HSID	Aoz980VxNz5W3NV7	.google...	/	2024-1...	21	✓	✓	None		High
	__Secure-3PSID	cwikQViOOPx_rksT0NWGqSEFyJgxZl6Dk1f6uRpj6...	.google...	/	2024-1...	85	✓	✓	None		High
www.jumbocash.n	SID	cwikQViOOPx_rksT0NWGqSEFyJgxZl6Dk1f6uRpj6...	.google...	/	2024-1...	74	✓	✓	None		High
ote tokens	__Secure-1PSID	cwikQViOOPx_rksT0NWGqSEFyJgxZl6Dk1f6uRpj6...	.google...	/	2024-1...	85	✓	✓	None		High
coups	APISID	KnNlpVCgjhneDFZl/ACggc8U1Fsfxby_C	.google...	/	2024-1...	40	✓	✓	None		High
rage	_GRECAPTCHA	09ALyjir-ogRoTMssqqv4GvZR6dzckAcEfQbfIRNjB0...	www.g...	/recapt...	2023-1...	100	✓	✓	None		High
rage	__Secure-1PSIDCC	ACA-Ox6CTAVmOagLtceaGxLcOP-ublX34J0Btlb...	.google...	/	2024-1...	91	✓	✓	None		High
ces	SIDCC	ACA-OxPwzStgf54rUTfHs75bjmEa84QVASJNb...	.google...	/	2024-1...	81	✓	✓	None		High
ard cache	atrium_connect	9b2807e147196a040595840c661441c5	www.ju...	/	2023-1...	46	✓	✓	None		Medium
nd fetch	jsa_session	9b2807e147196a040595840c661441c5	www.ju...	/	2023-1...	43	✓	✓	None		Medium
nd sync	NID	511=KxQ4ozNMQ3NfgZN2EHQYweKUX8LA7Q91j4...	.google...	/	2024-0...	705	✓	✓	None		Medium
acking mitigations	1P_JAR	2023-11-20-18	.google...	/	2023-1...	19	✓	✓	None		Medium
ns	__Secure-ENID	12.SE=WF37sQwHyArV2nIVP8JwuQ5DgPokFFbU7...	.google...	/	2024-0...	281	✓	✓	Lax		Medium
andler	UULE	a+cmt9sZTogMQpwcwm9kdWNlcjogMTkDgltZXNOY...	www.g...	/	2023-1...	198	✓	✓	None		Medium
background sync	AEC	Ackid1QDWwi170hNLHVNXKpC5h66JQWvAIcVU...	.google...	/	2024-0...	61	✓	✓	Lax		Medium
saging	OTZ	7293390_76_76_104100_72_446760	www.g...	/	2023-1...	33	✓	✓	None		Medium
API	usprivacy	1YNN	.wordp...	/	2024-1...	13	✓	✓	None		Medium
	S	billing-ui-v3=jdmupEhteNSELSSUVrNb_MSFoWtAx...	.google...	/Session	98	✓	✓	✓	None		Low
n rules											

The current *skey* can be found in the "atrium\_connect" and "jsa\_session" sections. I tried changing those values for a new key and replacing the URL *skey*, but got "logged out". I'm hesitant to say that this dynamic key security is foolproof, but as of right now, there haven't been any signs of easy exploitation methods.