

Task 1

The screenshot shows the OWASP ZAP interface. At the top, there are tabs for 'Site map', 'Scope', and 'Issues'. The 'Site map' tab is selected, showing a tree view of scanned hosts: '127.0.0.1:3000' (blue), '127.0.0.1:4200' (red), and 'localhost:3000' (green). Below the tree is a table of network requests:

Host	Method	URL	Params	Status code	Length	MIME type	Title
http://127.0.0.1:3000	GET	/socket.io/?EIO=4&tra...	✓	101	129		
http://127.0.0.1:3000	GET	/		200	75524	HTML	OWASP Juice Shop
http://127.0.0.1:3000	GET	/api/Quantitys/		200	6646	JSON	
http://127.0.0.1:3000	GET	/assets/i18n/en.json		200	33569	JSON	
http://127.0.0.1:3000	GET	/main.js		200	450720	script	
http://127.0.0.1:3000	GET	/polyfills.js		200	35325	script	
http://127.0.0.1:3000	GET	/rest/admin/applicati...		200	22119	JSON	
http://127.0.0.1:3000	GET	/rest/admin/applicati...		200	404	JSON	
http://127.0.0.1:3000	GET	/rest/products/search...	✓	200	14033	JSON	
http://127.0.0.1:3000	GET	/runtime.js		200	3818	script	
http://127.0.0.1:3000	GET	/socket.io/?EIO=4&tra...	✓	200	326	JSON	
http://127.0.0.1:3000	POST	/socket.io/?EIO=4&tra...	✓	200	215	text	
http://127.0.0.1:3000	GET	/socket.io/?EIO=4&tra...	✓	200	262	JSON	
http://127.0.0.1:3000	GET	/socket.io/?EIO=4&tra...	✓	200	230	text	
http://127.0.0.1:3000	GET	/vendor.js		200	1692893	script	
http://127.0.0.1:3000	GET	/api/Challenges/?nam...	✓	304	305	JSON	
http://127.0.0.1:3000	GET	/api/Challenges/					
http://127.0.0.1:3000	GET	/rest/products/search					
http://127.0.0.1:3000	GET	/socket.io/					

Below the table, there's a 'Request' and 'Response' detail view. The 'Request' tab is selected, showing a 'Pretty' dump of the request:

```
1 GET /socket.io/?EIO=4&transport=websocket&sid=AeA6sxIHolszDw7BAAAc HTTP/1.1
2 Host: 127.0.0.1:3000
3 Connection: Upgrade
4 Pragma: no-cache
5 Cache-Control: no-cache
6 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/144.0.0.0 Safari/537.36
7 Upgrade: websocket
8 Origin: http://127.0.0.1:3000
9 Sec-WebSocket-Version: 13
10 Accept-Encoding: gzip, deflate, br
11 Accept-Language: en-US, en; q=0.9
12 Sec-WebSocket-Key:
4wbwrOP6azkhJaAxvsDpJQ==
```

The 'Response' tab is also visible. To the right, there's an 'Inspector' panel with sections for Request attributes, Request query parameters, Request headers, and Response headers. A 'Notes' section is also present.

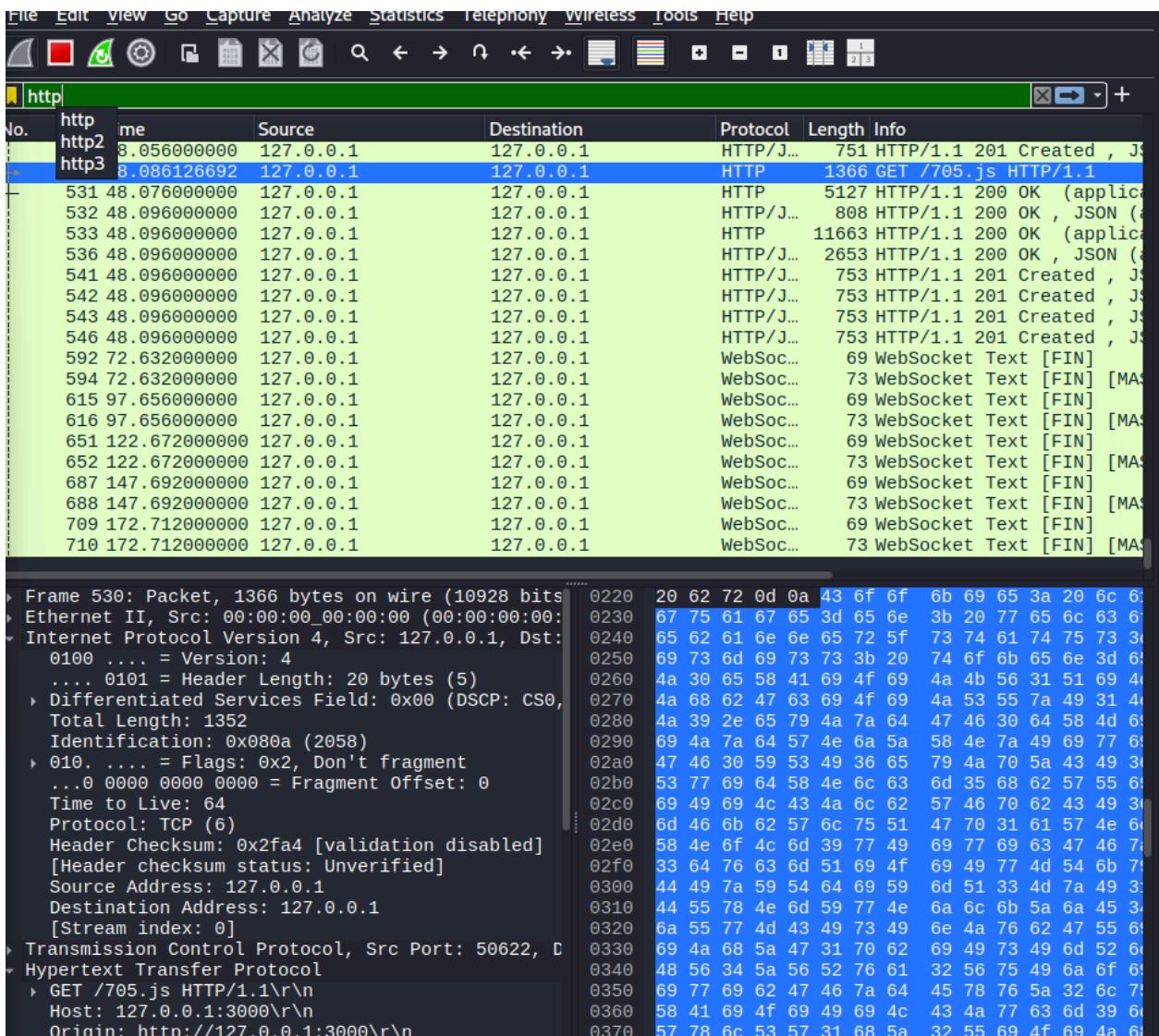
Hidden paths can tell an attacker what services exist, application architecture, technologies used and give information on where to focus an attack. Authentication exists but APIs may not enforce it properly, some endpoints (unused or outdated/old) may skip auth check

Task 2

The screenshot shows a web browser window for the OWASP Juice Shop application. The URL is 127.0.0.1:3000/#/search. The page displays a sidebar menu with sections: Account, Contact, and Company. Under Account, there are links for Orders & Payment (with a checked checkbox), Privacy & Security, and Logout. Under Contact, there are links for Customer Feedback, Complaint, and Support Chat. Under Company, there are links for About Us, Photo Wall, and Deluxe Membership. The main content area is a search results page titled 'Items per page: 12' with '0 of 0' results. The top right corner shows a shopping cart icon with '0' items.

The input '`' OR 1=1 --`' alters the SQL logic by forcing the authentication condition to always evaluate as true and commenting out the remaining checks. As a result, the database returns a user record regardless of credentials, causing the application to treat the login as successful.

Task 3



Without HTTPS, attackers could extract credentials, session cookies, personal data, and application endpoints from plaintext HTTP traffic, enabling account hijacking, privacy breaches, and further targeted attacks.

Reflection

The vulnerabilities demonstrated in this lab closely reflect techniques used by real-world cyberattackers. Information disclosure through exposed paths and metadata allows attackers to map applications and identify weak points without authentication, often serves as the first step in an attack chain. SQL injection demonstrates how improper input handling can directly compromise authentication mechanisms and lead

to unauthorized access or data breaches. Observing plaintext network traffic highlights how lack of encryption enables credential theft and session hijacking exploited together rather than in isolation, increasing the overall impact and potential danger of an attack