

Task 1

The screenshot displays the Burp Suite interface. The top section shows the 'Site map' tab with a filter: 'Site map filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders'. The site map lists three hosts: 127.0.0.1:3000, 127.0.0.1:4200, and localhost:3000. The 127.0.0.1:3000 host is expanded, showing a list of HTTP 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/					

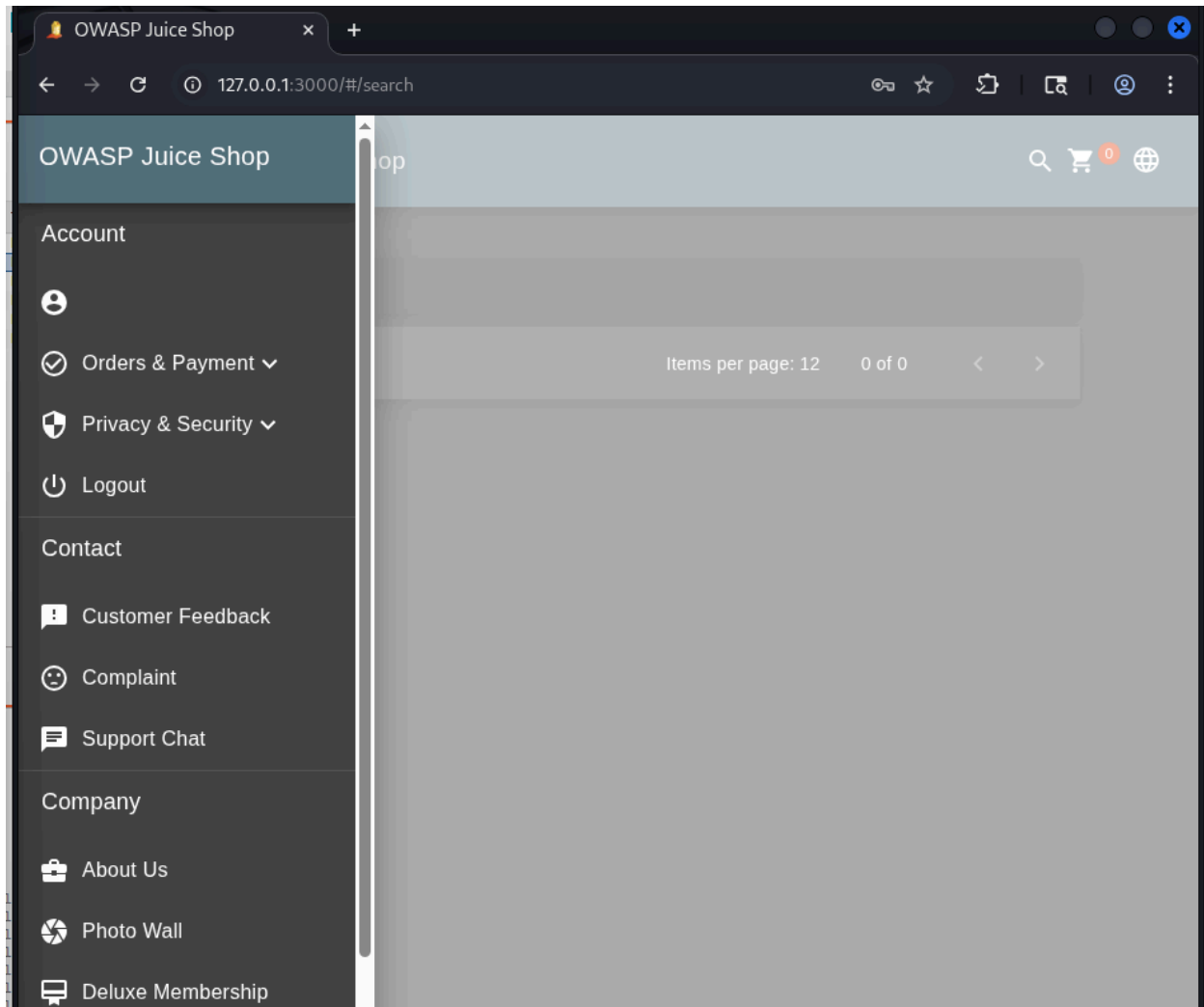
The bottom section shows the 'Request' and 'Response' tabs. The 'Request' tab is selected, showing a 'Pretty' view of the request. The request is a GET request to /socket.io/?EIO=4&transport=websocket&sid=AeA6sxIHolzDw7BAAC HTTP/1.1. The request includes headers: Host: 127.0.0.1:3000, Connection: Upgrade, Pragma: no-cache, Cache-Control: no-cache, User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/144.0.0.0 Safari/537.36, Upgrade: websocket, Origin: http://127.0.0.1:3000, Sec-WebSocket-Version: 13, Accept-Encoding: gzip, deflate, br, Accept-Language: en-US,en;q=0.9, Sec-WebSocket-Key: 4wbwrOP6azkhJaAxsVdpJQ==.

The 'Inspector' tab is also visible, showing 'Request attributes' (2), 'Request query parameters' (3), 'Request headers' (11), and 'Response headers' (3).

The bottom status bar shows 'Event log (6)', 'All issues', 'Memory: 130.8MB of 980.0MB', and 'Disabled'.

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 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

The image displays a Wireshark network traffic capture. The top pane shows a list of captured packets, with packet 531 selected. The middle pane shows the details of the selected packet, which is an HTTP GET request for /705.js. The bottom pane shows the raw packet data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
531	48.076000000	127.0.0.1	127.0.0.1	HTTP	1366	GET /705.js HTTP/1.1
532	48.096000000	127.0.0.1	127.0.0.1	HTTP	5127	HTTP/1.1 200 OK (application/javascript)
533	48.096000000	127.0.0.1	127.0.0.1	HTTP	808	HTTP/1.1 200 OK (application/javascript)
536	48.096000000	127.0.0.1	127.0.0.1	HTTP	11663	HTTP/1.1 200 OK (application/javascript)
541	48.096000000	127.0.0.1	127.0.0.1	HTTP	2653	HTTP/1.1 200 OK, JSON (application/javascript)
542	48.096000000	127.0.0.1	127.0.0.1	HTTP	753	HTTP/1.1 201 Created (application/javascript)
543	48.096000000	127.0.0.1	127.0.0.1	HTTP	753	HTTP/1.1 201 Created (application/javascript)
546	48.096000000	127.0.0.1	127.0.0.1	HTTP	753	HTTP/1.1 201 Created (application/javascript)
592	72.632000000	127.0.0.1	127.0.0.1	WebSocket	69	WebSocket Text [FIN]
594	72.632000000	127.0.0.1	127.0.0.1	WebSocket	73	WebSocket Text [FIN]
615	97.656000000	127.0.0.1	127.0.0.1	WebSocket	69	WebSocket Text [FIN]
616	97.656000000	127.0.0.1	127.0.0.1	WebSocket	73	WebSocket Text [FIN]
651	122.672000000	127.0.0.1	127.0.0.1	WebSocket	69	WebSocket Text [FIN]
652	122.672000000	127.0.0.1	127.0.0.1	WebSocket	73	WebSocket Text [FIN]
687	147.692000000	127.0.0.1	127.0.0.1	WebSocket	69	WebSocket Text [FIN]
688	147.692000000	127.0.0.1	127.0.0.1	WebSocket	73	WebSocket Text [FIN]
709	172.712000000	127.0.0.1	127.0.0.1	WebSocket	69	WebSocket Text [FIN]
710	172.712000000	127.0.0.1	127.0.0.1	WebSocket	73	WebSocket Text [FIN]

Frame 530: Packet, 1366 bytes on wire (10928 bits)
Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00)
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
... = Version: 4
... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length: 1352
Identification: 0x080a (2058)
... = Flags: 0x2, Don't fragment
... 0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 64
Protocol: TCP (6)
Header Checksum: 0x2fa4 [validation disabled]
[Header checksum status: Unverified]
Source Address: 127.0.0.1
Destination Address: 127.0.0.1
[Stream index: 0]
Transmission Control Protocol, Src Port: 50622, Dst Port: 50622
Hypertext Transfer Protocol
GET /705.js HTTP/1.1
Host: 127.0.0.1:3000
Origin: http://127.0.0.1:3000

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