SCENARIO

The lab's "Check stock" functionality accepts XML input and reflects unexpected values back in its responses. Notably, there exists a simulated EC2 metadata endpoint within the server.

**PROCEDURE**

1. We access a product page and click "Check stock". Simultaneously, we use Burp Suite to capture the POST request.
2. Introduce this external entity definition between the XML declaration and the stockCheck element: <!DOCTYPE test [ <!ENTITY xxe SYSTEM "http://169.254.169.254/"> ]>
3. Update the productId number using &xxe;. The server response should then exhibit "Invalid product ID:" and reflect the metadata endpoint's initial folder name.
4. Subsequently, fine-tune the DTD URL to navigate through the API until reaching /latest/meta-data/iam/security-credentials/admin, which will divulge the SecretAccessKey within a JSON structure.

**PAYLOAD**

<!DOCTYPE test [ <!ENTITY xxe SYSTEM "http://169.254.169.254/"> ]>

**PROOF OF CONCEPT**

**REMEDIATION**

1. **Outbound Restrictions:** Configure the application server to restrict outbound connections, ensuring only necessary communications are allowed.
2. **Robust Input Validation:** Incorporate thorough input validation mechanisms and enforce output encoding to prevent unexpected data from being processed.