Finding Name: Session Hijacking - Insufficient Session Binding

Name	Team	Role	Project	Quality Assurance	Is this a re-tested Finding?
Andrew Duffy	AppAtack	Pen-Tester	OnTrack Web App	Filipe Oliveira	No

Was this Finding Successful?
Yes

Finding Description

A Session Hijacking vulnerability was discovered in the OnTrack web application as a result of insufficient session binding. Session binding is the process of associating a session token with specific client attributes such as IP address, user's browser, or a device fingerprint. Without strong session binding, a valid session token can be reused across different devices or locations, which increases the risk of session hijacking if a token is intercepted or stolen.

During testing, a valid authentication token and username captured using Burp Suite, from an active privileged user session (convenor) were inserted into an intercepted request made by a lower-privileged user (student). The request was to carry out privileged user actions and is blocked by default for the low-level user. However, replacing the low-level token with the privileged token and forwarding the request, it was accepted and processed successfully by the application.

This vulnerability demonstrated that no additional validation was in place to bind the session token to the original user's environment. This vulnerability can be exploited by an attacker who successfully gains access to an active session token (e.g. via Cross Site Scripting (XSS), social engineering, or sniffing). Insufficient session binding provides an opportunity for threat actors to reuse the stolen token from another browser or device and impersonate the original user. During testing the stolen token enabled the low-level user to conduct privileged actions but was unable to take over the privileged session at this time.

Due to the common nature of this vulnerability, it should be expected that threat actors will attempt to search for and exploit it with a Moderate likelihood of occurring. Exploitation allowed a threat actor to perform some administrative actions but was not able to fully comprise the privileged session. It was also noted that compensating controls appeared to be in place, which included a time-based session token rotation which greatly limits the window of opportunity for exploitation and reducing the impact to Significant, giving an overall Risk of Medium.

Risk Rating - Medium

Impact: **Significant**
Likelihood: **Moderate**

Impact values				
Very Minor	**Minor**	**Significant**	**Major**	**Severe**
Risk that holds little to no impact. Will not cause damage and regular activity can continue.	Risk that holds minor form of impact, but not significant enough to be of threat. Can cause some damage but not enough to impede regular activity.	Risk that holds enough impact to be somewhat of a threat. Will cause damage that can impede regular activity but will be able to run normally.	Risk that holds major impact to be of threat. Will cause damage that will impede regular activity and will not be able to run normally.	Risk that holds severe impact and is a threat. Will cause critical damage that can cease activity to be run.

Likelihood				
Rare	**Unlikely**	**Moderate**	**High**	**Certain**
Event may occur and/or if it did, it happens in specific circumstances.	Event could occur occasionally and/or could happen (at some point)	Event may occur and/or happens.	Event occurs at times and/or probably happens a lot.	Event is occurring now and/or happens frequently.

Business Impact

While the test did not demonstrate full session takeover, the results indicate a significant weakness in the application's session handling. The lack of session binding allows a valid session token to be reused across different browsers or devices simultaneously. If a session token is successfully intercepted or obtained by a threat actor, it can then be reused to impersonate the original user and perform any privileged actions available to the original user. Depending on the victim's token, this vulnerability presents a risk of unauthorised access to sensitive data or privileged functionality. Although time-based session expiry mechanisms may reduce the window of opportunity, the absence of session binding increases the risk of token reuse and may compromise the confidentiality and integrity of the data within the application.

Affected Assets

OnTrack Web Applicaiton Authentication/Session Token implementation. Any API, endpoint of functionality that relies on a valid authentication token is potentially vulnerable.

Evidence

Step 1: Set up separate browser instances

1. This requires two isolated browser contexts. one for unprivileged user (student_1), one for privileged user (aadmin). Start Burp Suite. Open Firefox and set traffic to be forwarded through the Burp proxy. This is one of the isolated browsers

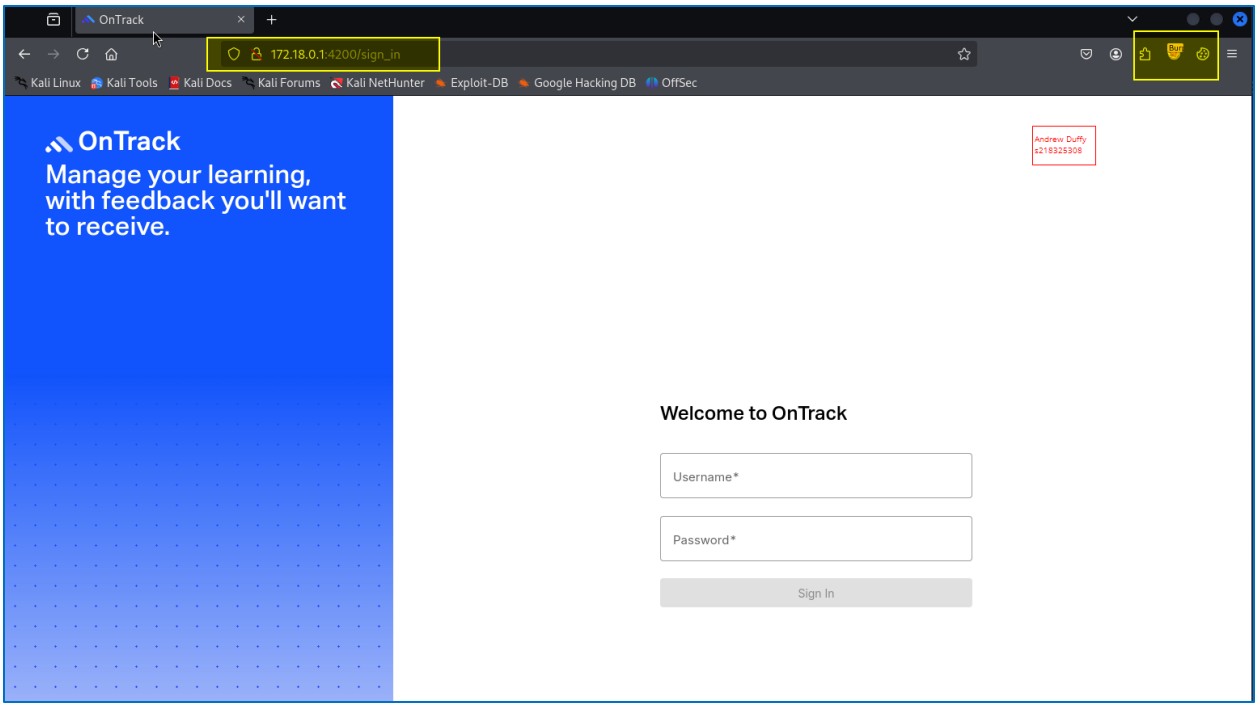

Figure – Browser 1, Firefox sending traffic through Burp Suite

2. Navigate to the proxy tab in Burp Suite and select open browser. This will open a chromium-based browser tied to Burp Suite. This will act as the second isolated browser instance.

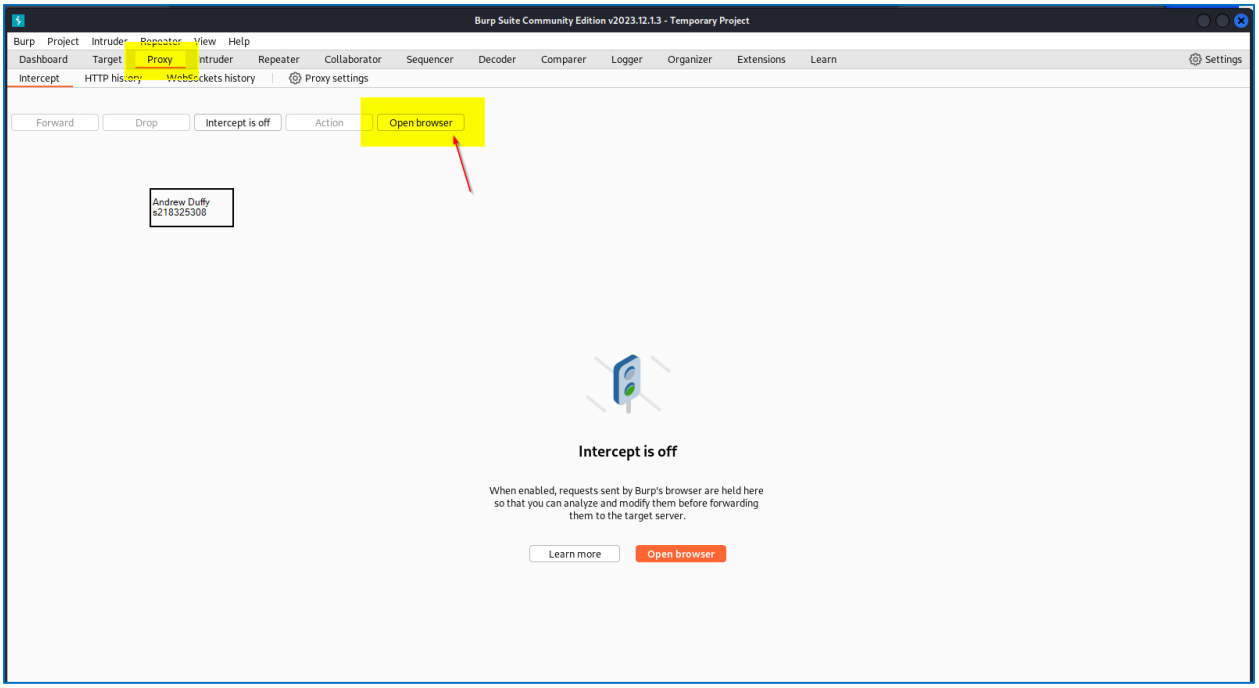

Figure – Open Burp inbuilt browser

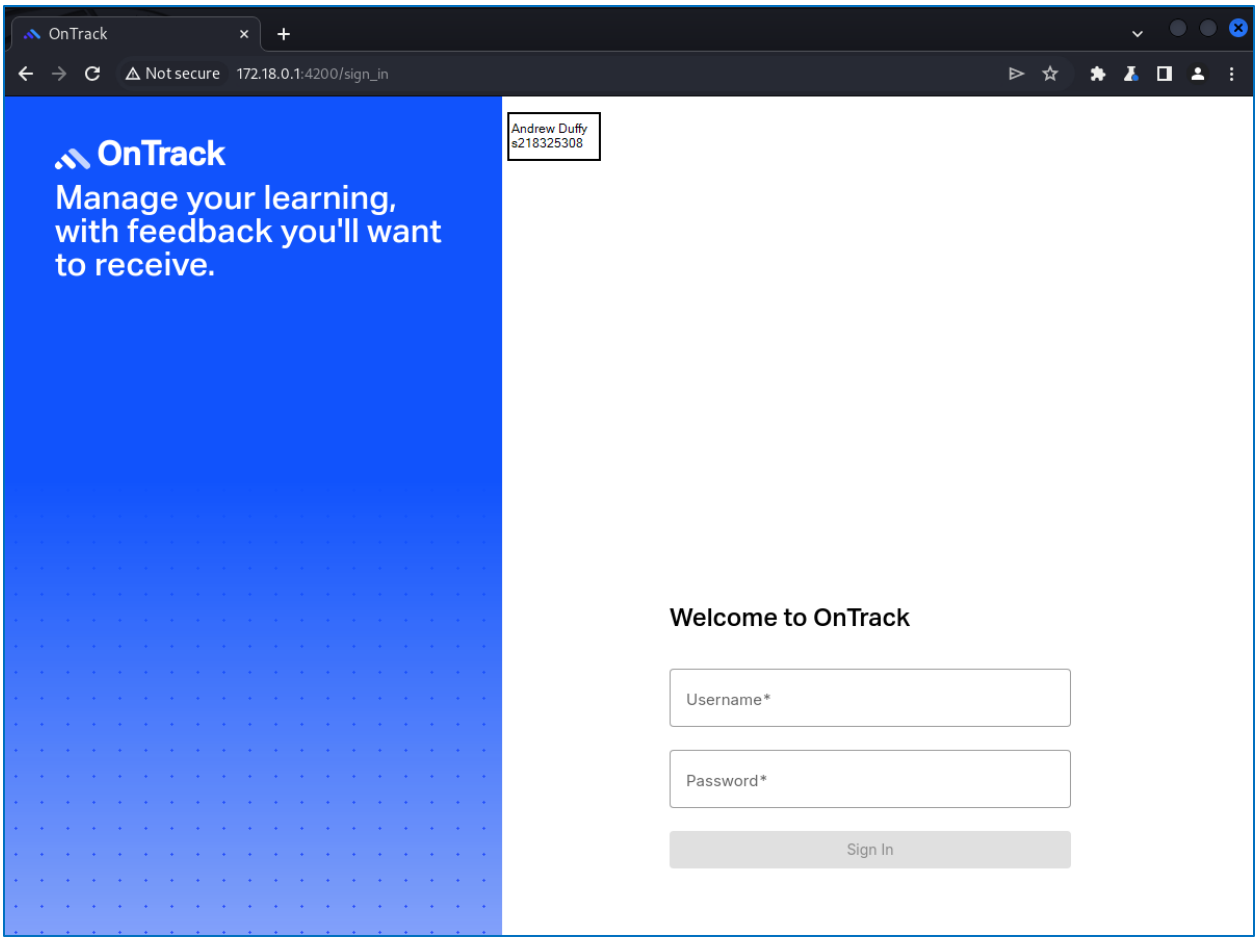

Figure – Burp Suite inbuilt browser

Step 2: Login to each account

1. Login to two separate accounts using the individual browsers. In the Firefox browser as student_1. This will navigate you to /home once authenticated.

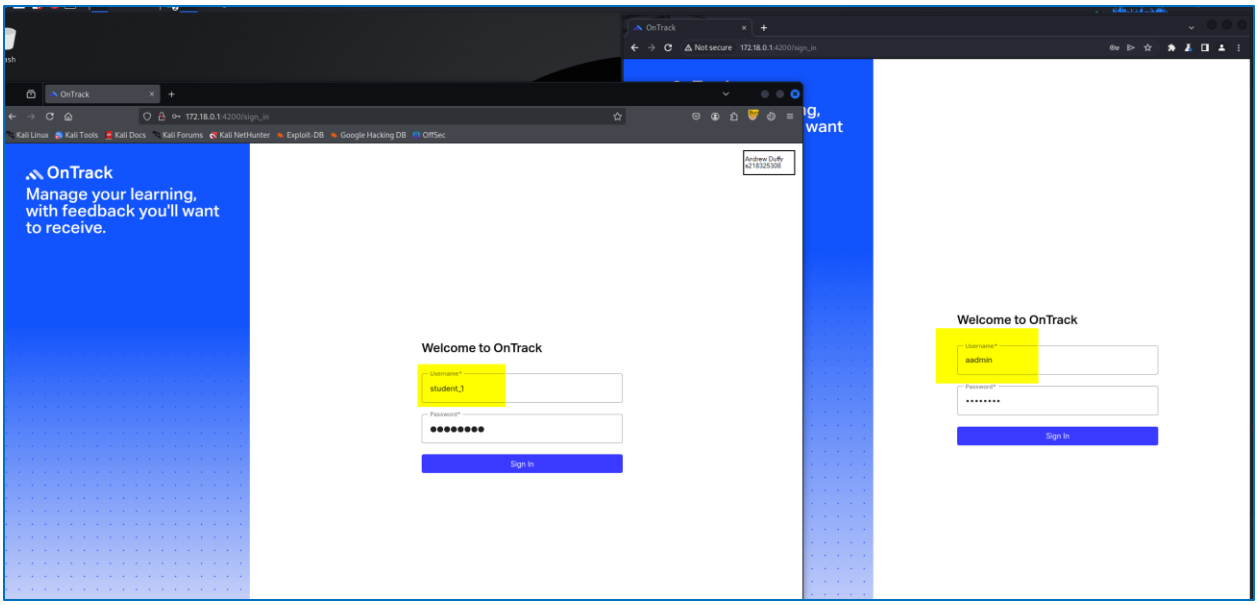

Figure – Set up both users

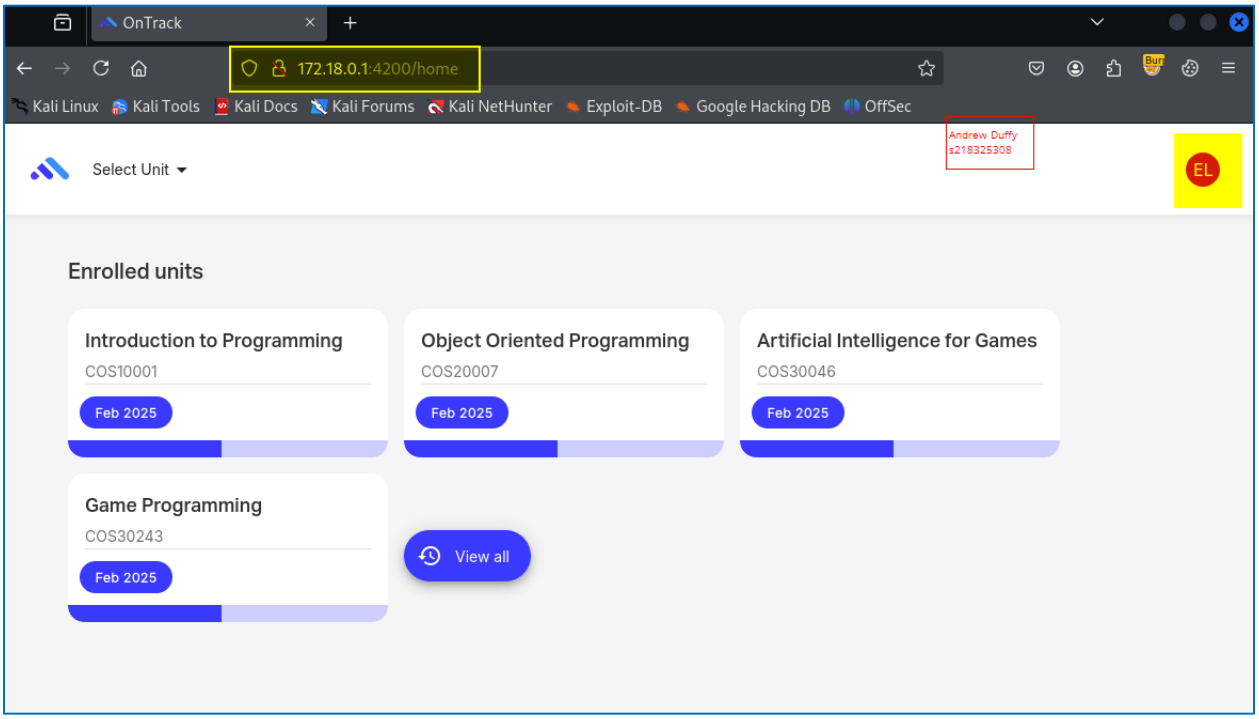

Figure – Logged in a student_1 a student level user.

2. Login to aadmin using the Burp browser. This will present the admin home page.

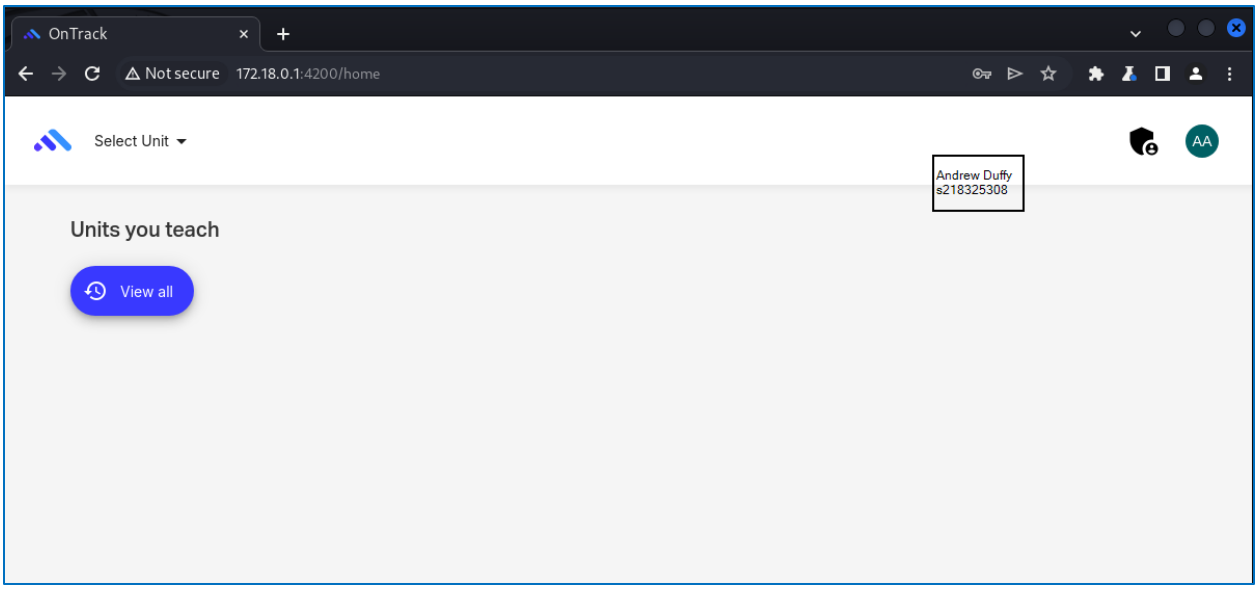

Figure – Logged in as aadmin in browser 2

Step 3: Emulate capturing privileged token

1. Open Burp Suite and review the HTTP history. Find the authorisation request and save the token that is provided in the response.

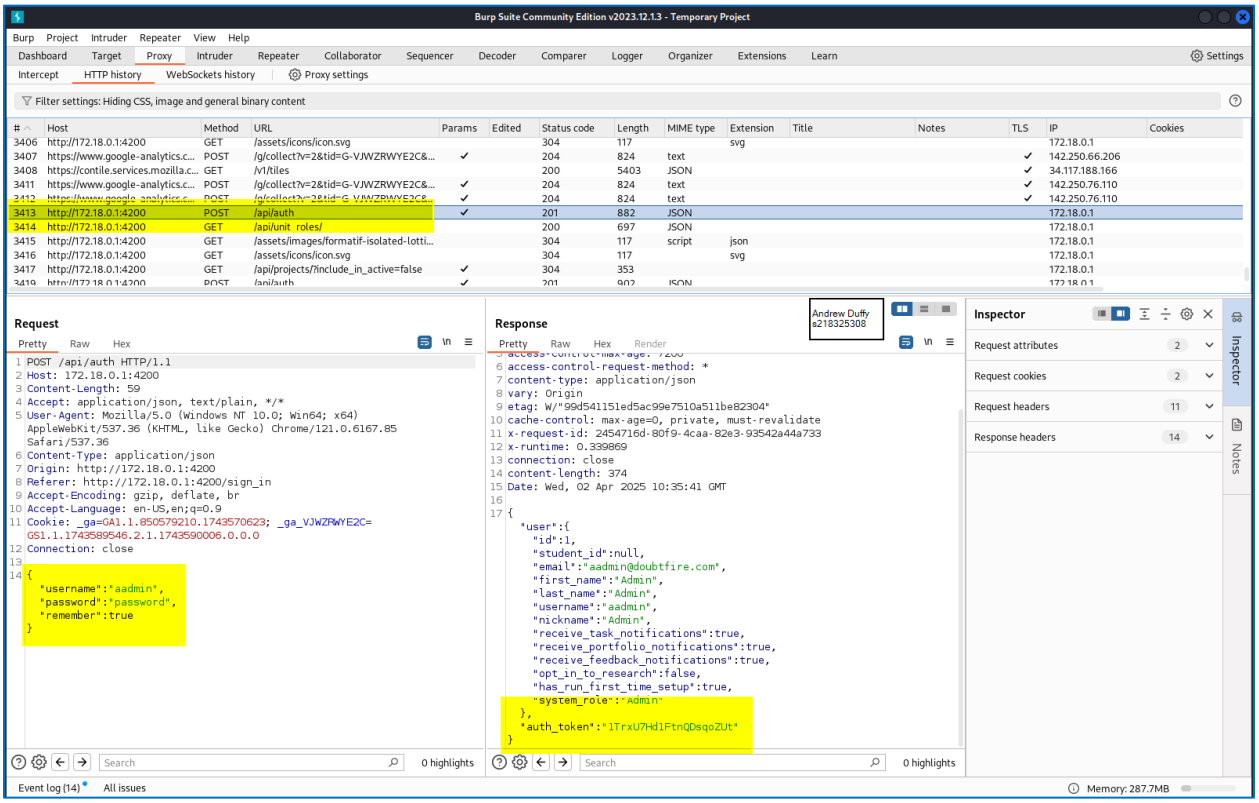

Figure – 'Obtaining' victim token

Step 4: Execute attack

1. For Demonstration - On the unprivileged account, using the Broken Access Control vulnerability, to navigate to the **/admin/units** endpoint. This is a privileged endpoint accessible due to a broken access control vulnerability.

2. Normally the low-level user cannot execute the privileged function of creating a new unit.

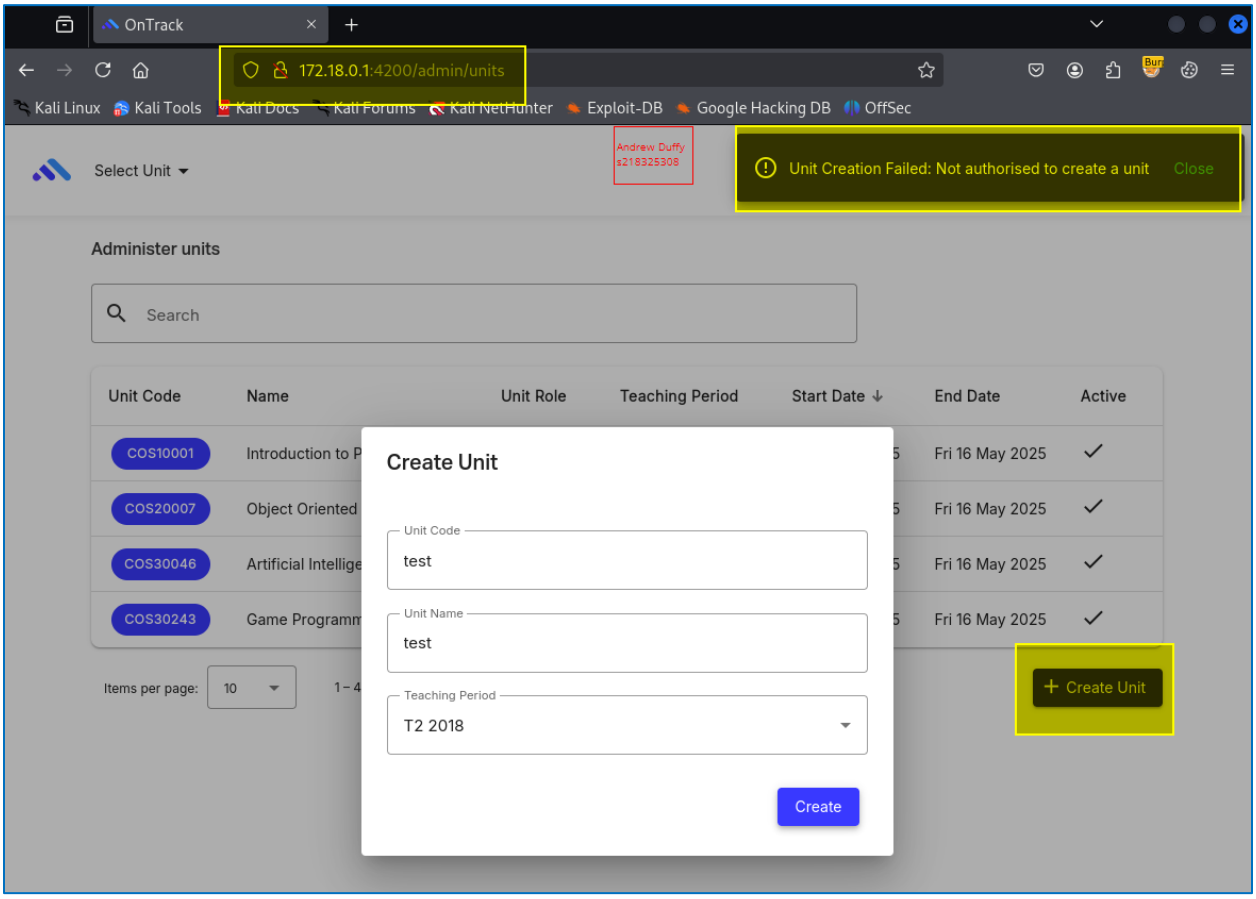

Figure – Student level user attempting to execute admin level function

3. Reattempt to create a new unit and intercept the request in Burp Suite. Complete the required details and turn on the proxy intercept.

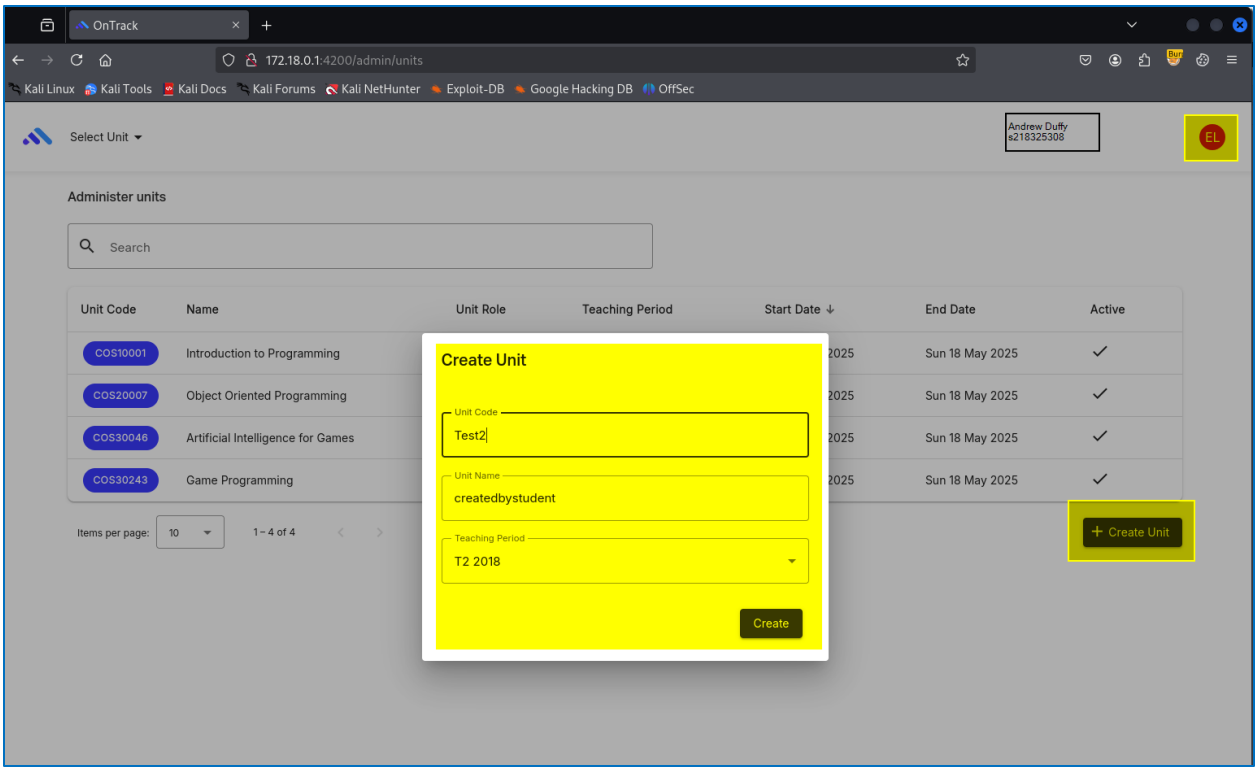

Figure – Fill out required details

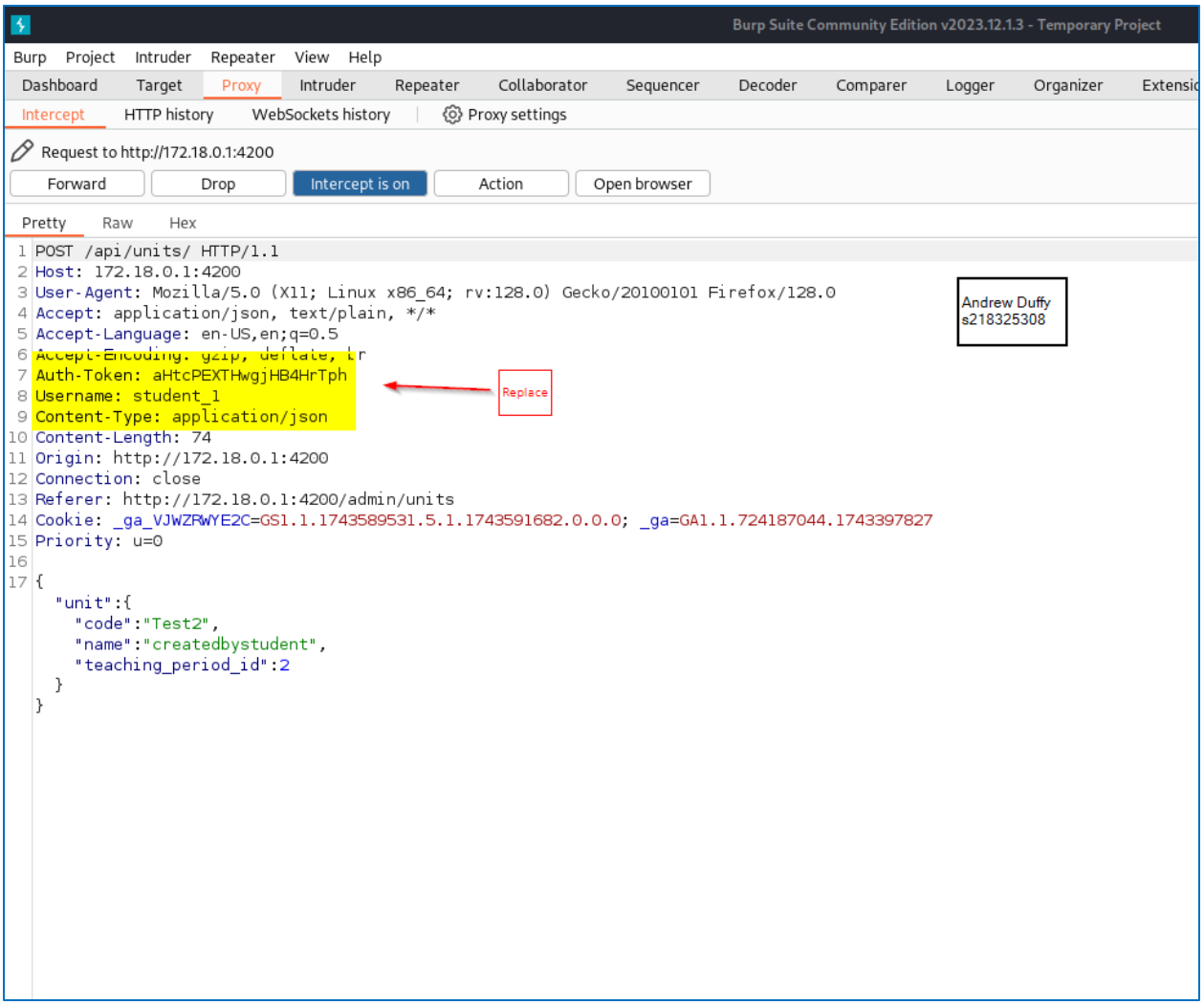

Figure – Review the intercepted request. Identify Authentication token and username.

4. Replace authentication token and username with ‘stolen’ token and username.

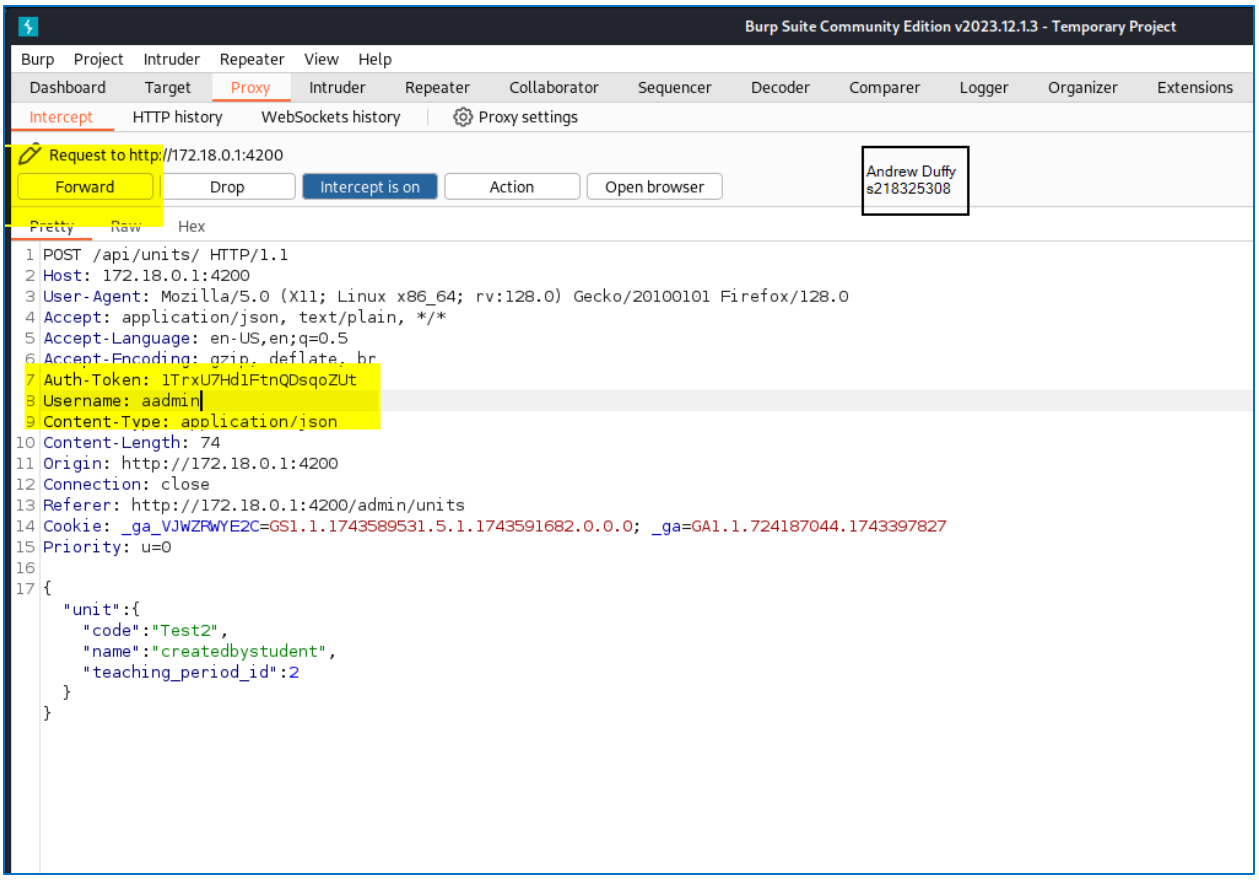

Figure – Fill out required details

5. Forward the request – this will successfully add the newly created unit, validating the session token can be used outside the original context.

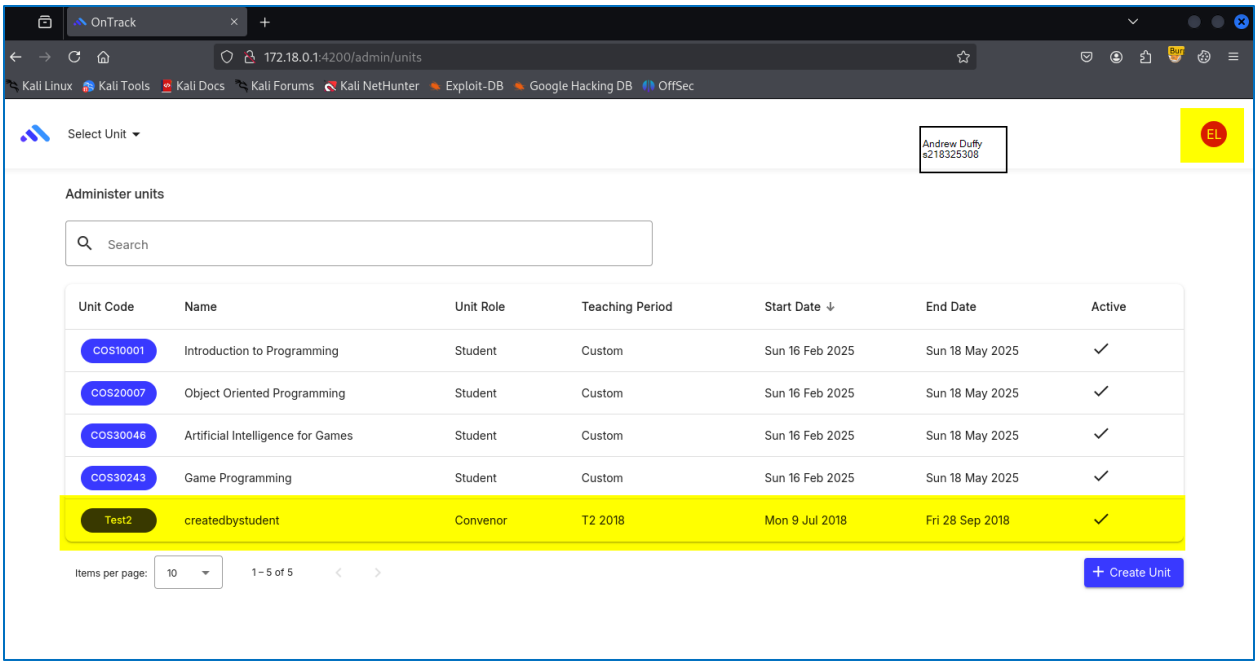

Figure – New unit added by low level user

Remediation Advice

The application does not enforce strong session binding between the client and session token, which allows for session token reuse from unauthorised environments.

Mitigations:

- Implement session binding mechanisms by associating session tokens with client-specific attributes (e.g. IP address, User-Agent, device fingerprint) and validating them on each request.
- Disable Web Browser Cross-Tab Sessions – once a user has logged in and a session has been established a user must re-authenticate if a new web browser tab or window is opened against the same web application.

References

- OWASP Top 10 – Broken Access Control
- OWASP Session Managment Cheat Sheet

Contact Details

Name/Teams: Andrew Duffy
Email: s218325308@deakin.edu.au

Pentest Leader Feedback.

Great find! Excellent work.
Filipe Oliveira S222478779@deakin.edu.au