Finding Name: Insecure Token Exposure via Client-Side Storage and HTTP Headers

Name	Team	Role	Project	Quality Assurance	Is this a re-tested Finding?
Filipe Oliveira	OnTrack	Pen-Tester	AppAttack	Darryl Ooi	No

Was this Finding Successful?
Yes

Finding Description

Upon successful login to the OnTrack platform, the application issues a persistent authentication token that is sent in every subsequent API request via HTTP headers. This token provides full user session access without needing credentials again. The application is served over unencrypted HTTP meaning this token is transmitted in plaintext and can be intercepted using tools such as Wireshark on the same network.

Risk Rating

Impact: Significant
Likelihood: High

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

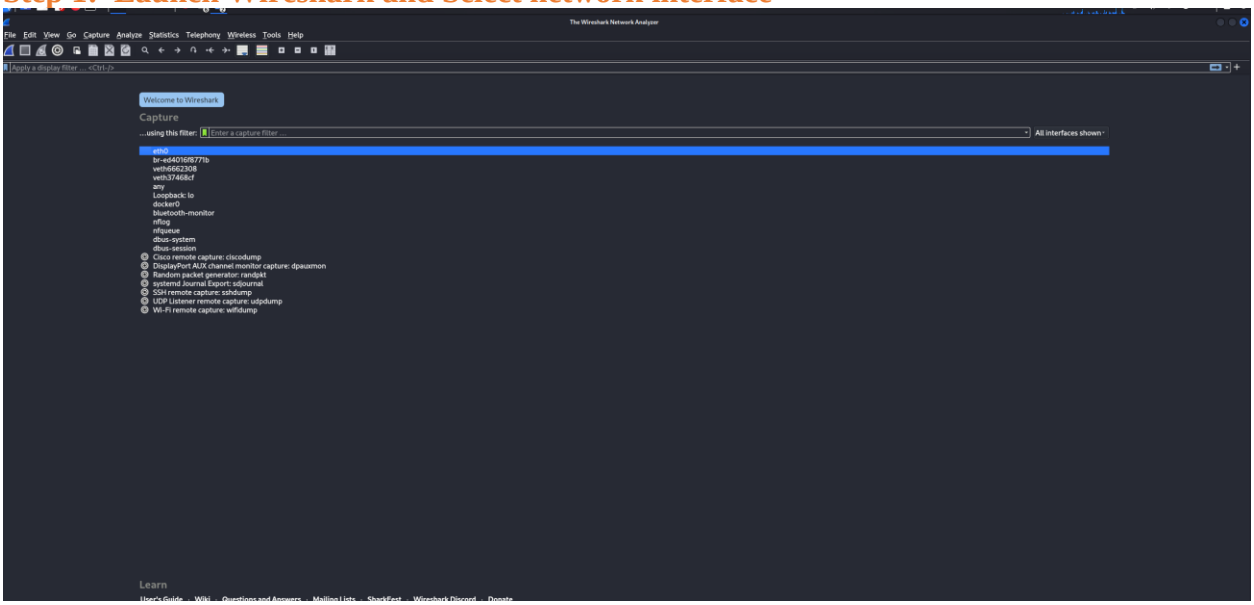
If an attacker gains access to a user's authentication token, they can fully impersonate the victim without needing a password. This could lead to exposure of academic records, assignment submissions and may allow unauthorised access to admin modifications. If this occurred on a live, production grade system over an open or shared network, it could result in unauthorised access to student information, data privacy violations and erosion of trust in the system's security.

Affected Assets

- Ontrack web application
- API endpoints using Authentication tokens
- Student session tokens.

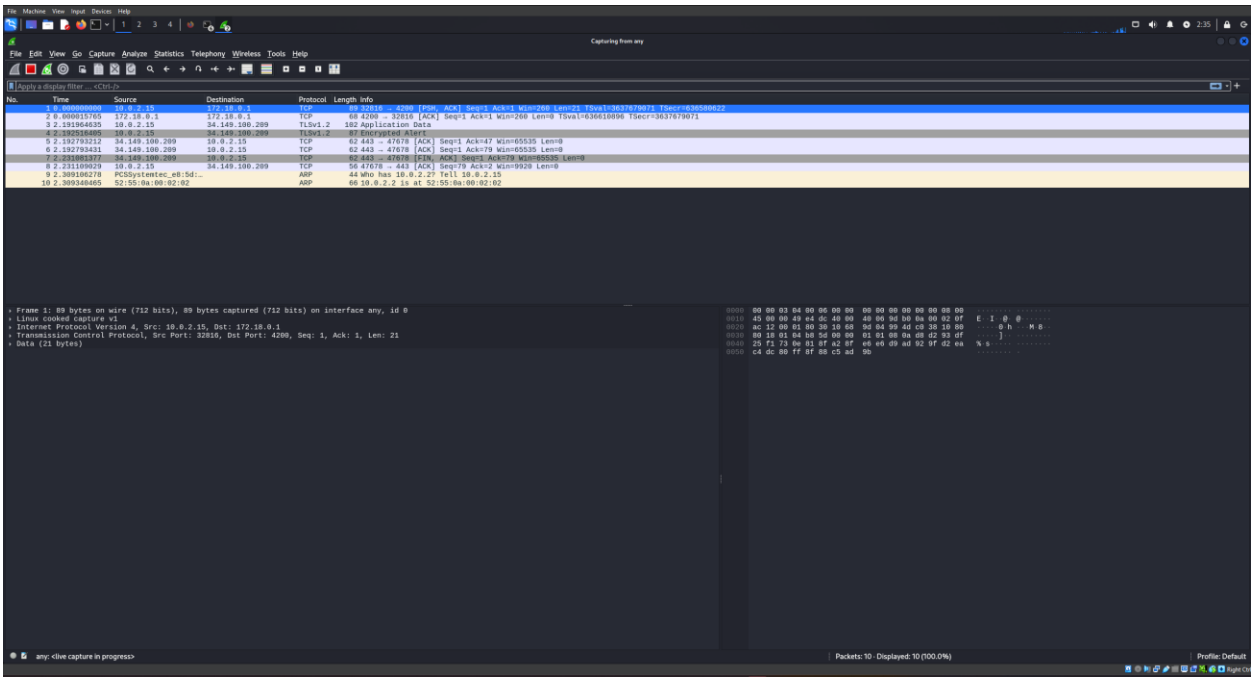
Evidence

Step 1: Launch Wireshark and Select network interface



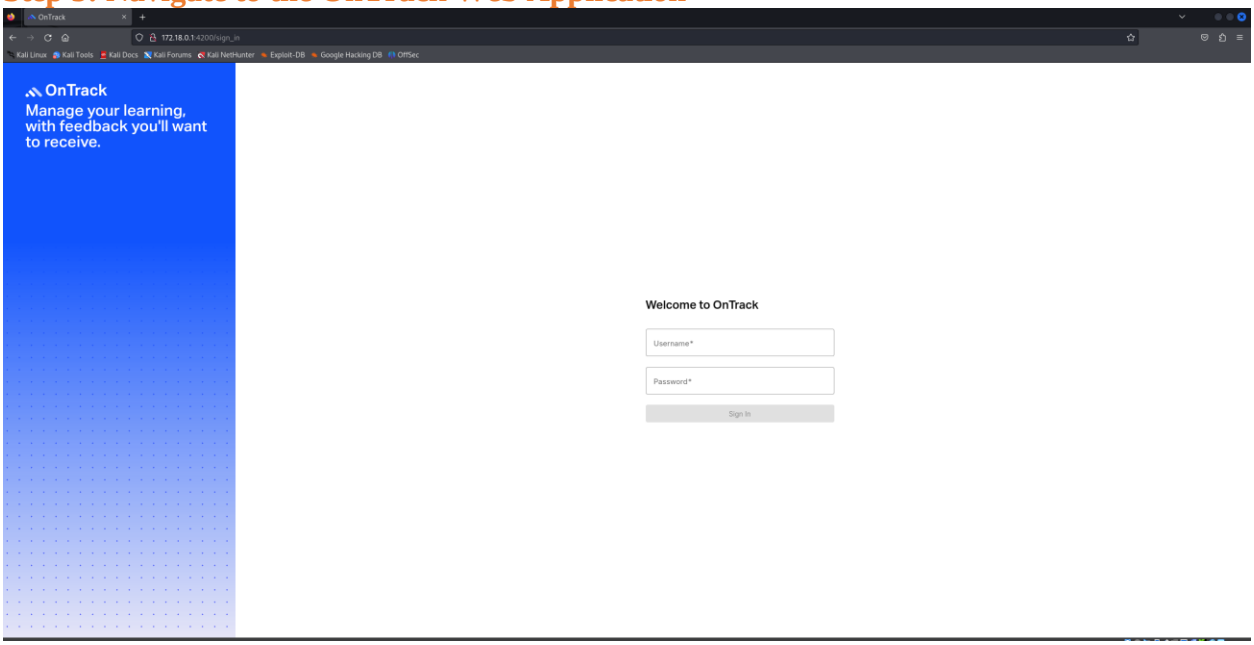
Open Wireshark and select the interface labeled" any" to monitor all the network traffic.

Step 2: begin capturing packets

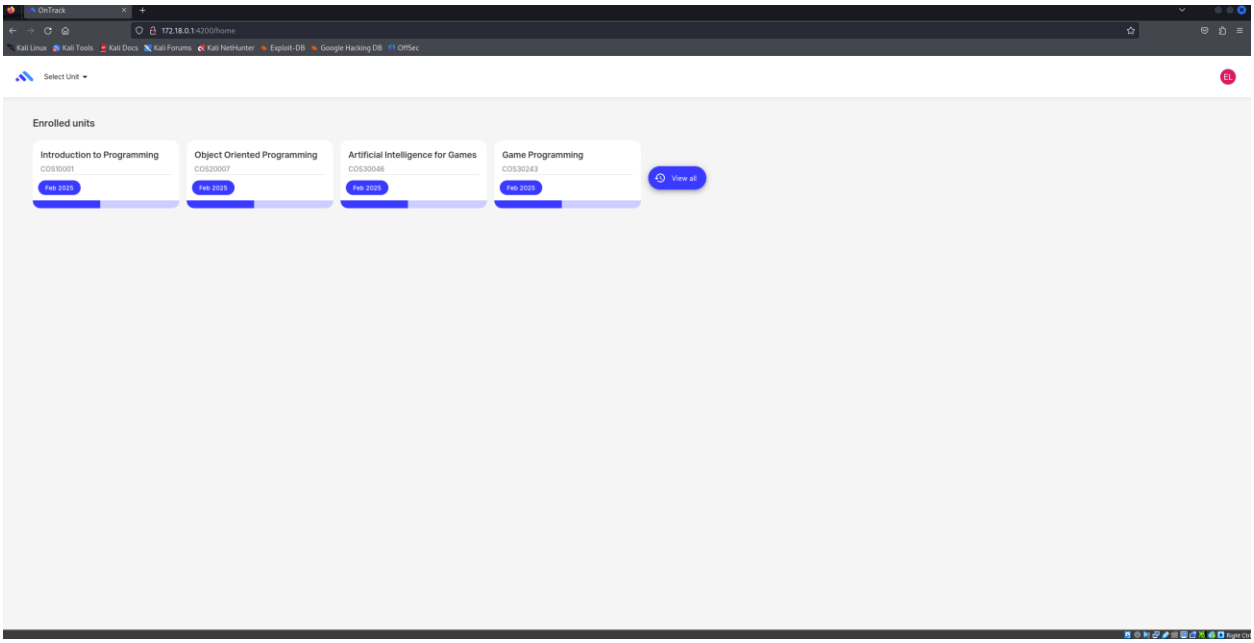


Click the blue fin under the top left File menu to begin live packet capture.

Step 3: Navigate to the OnTrack Web Application

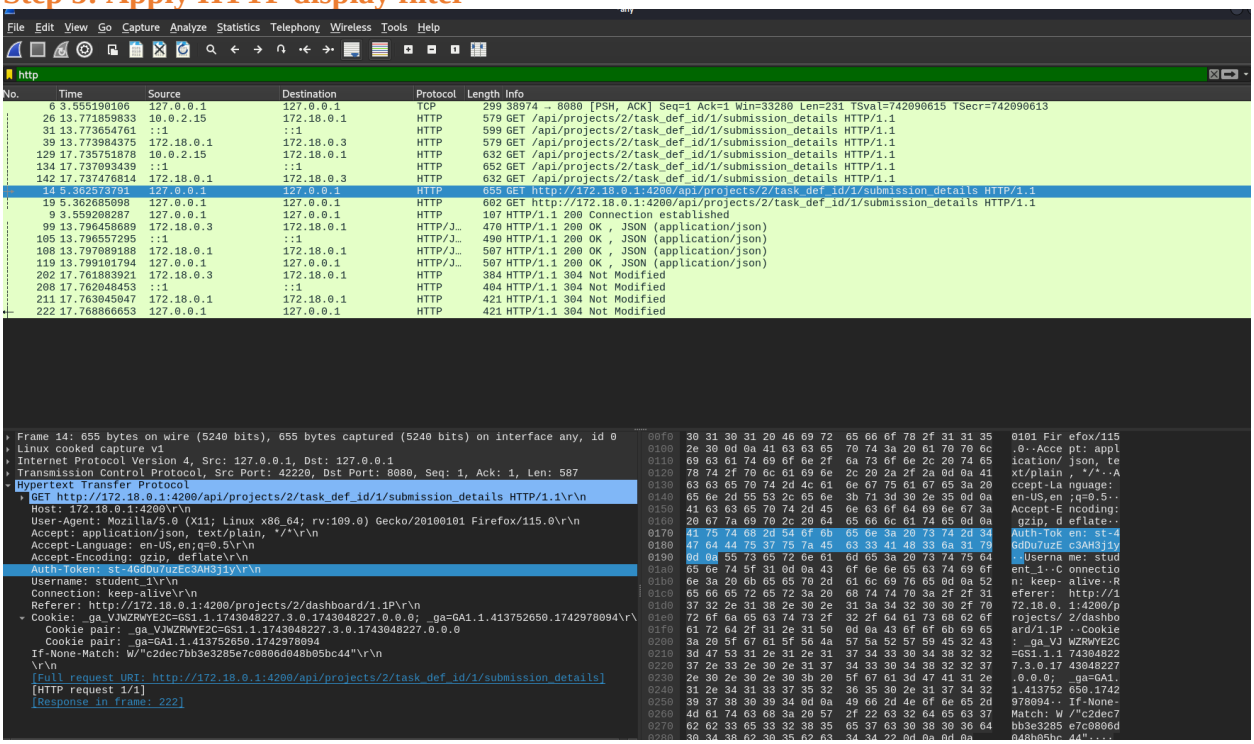


Step 4: Login with student credentials



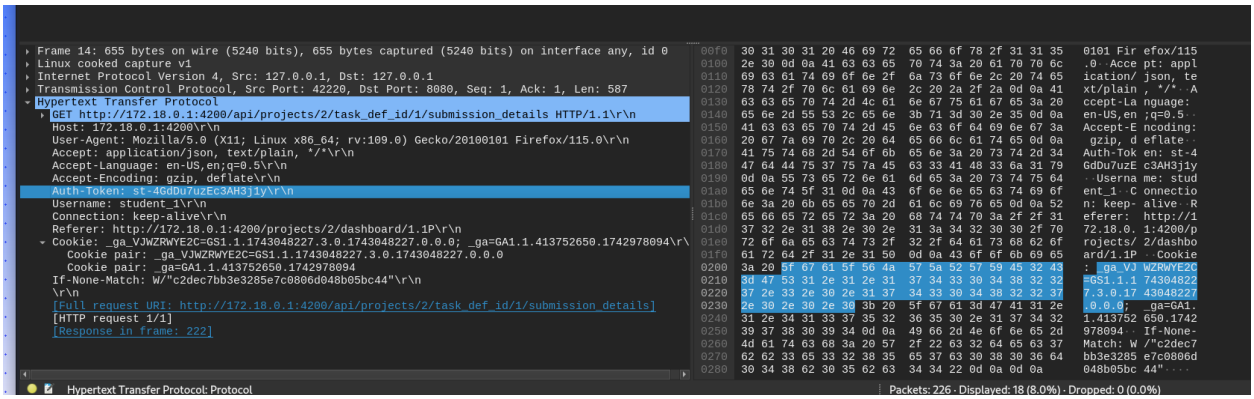
Login with student_1 and password you can also do anything on ontrack, like clicking on a subject.

Step 5: Apply HTTP display filter



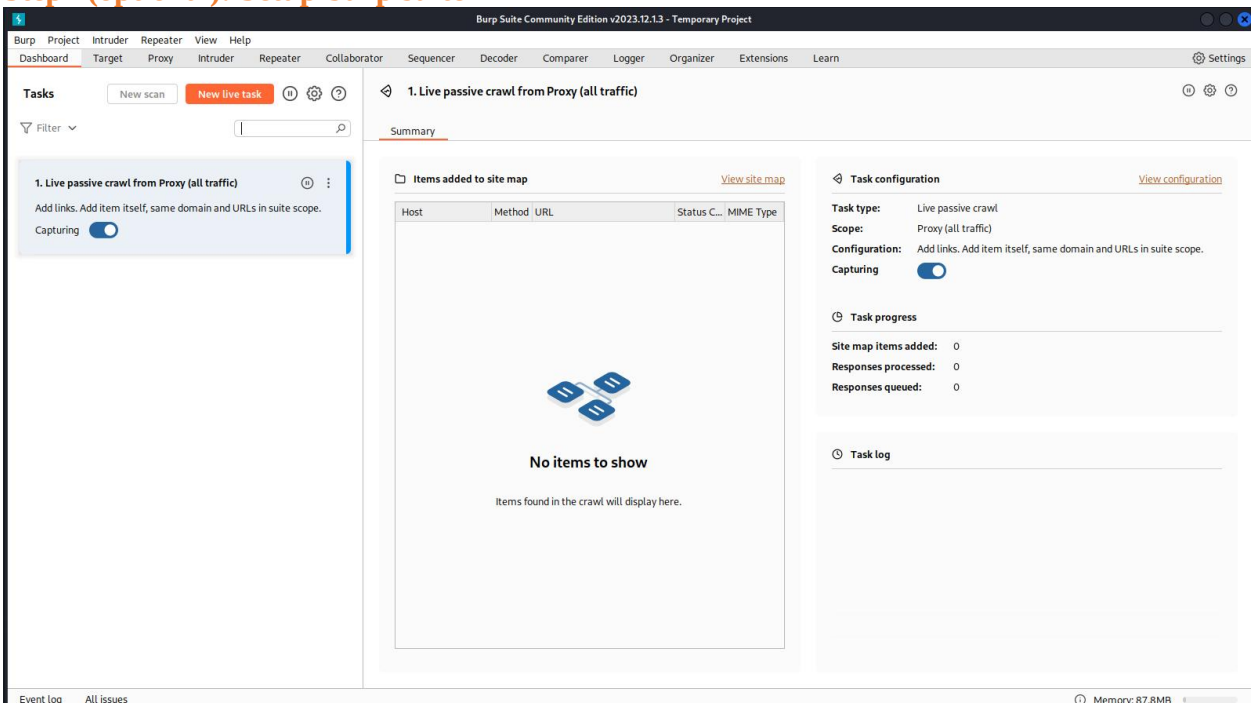
Type http in the filter bar at the top

Step 6: Locate and inspect a Request Packet.

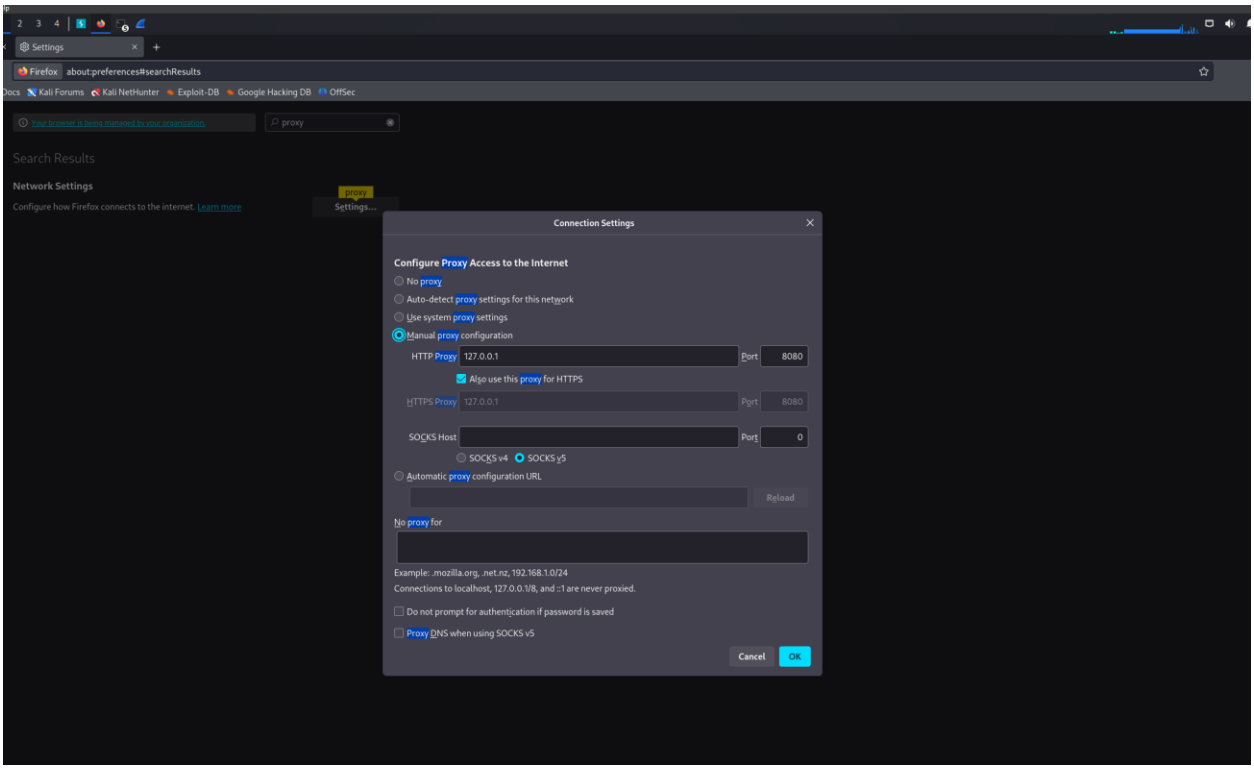


Click on any packet and find the Hypertext protocol subheading and click expand, if it is a packet that has been sent after your login it will contain cookie session data and the authentication token for the login Aswell as the username

Step 7(optional): Setup burp suite

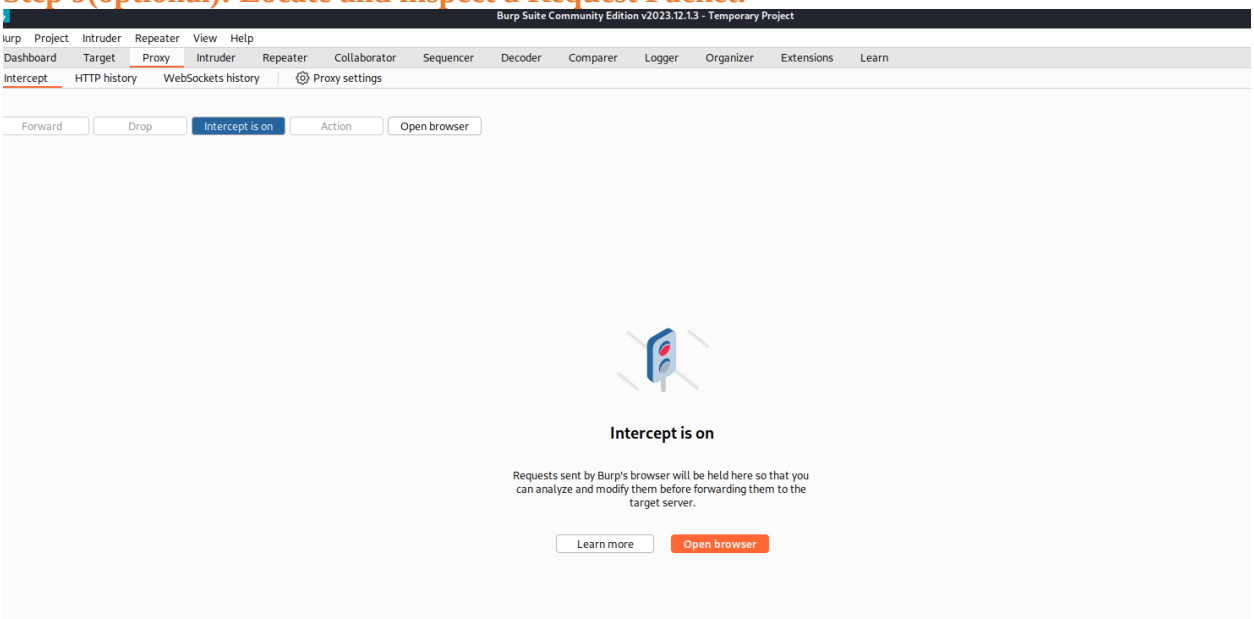


Step 8 (optional): Setup proxy to burp



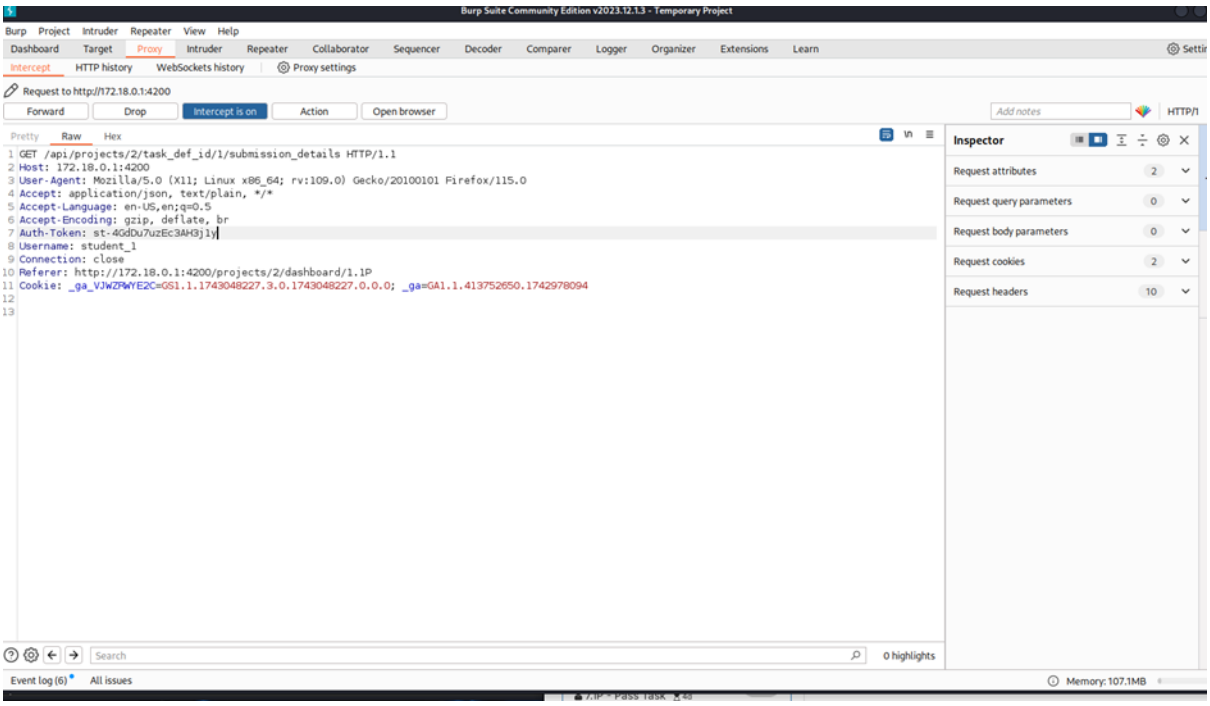
Go to settings in your browser and turn the proxy and input the proxy ip and port

Step 9(optional): Locate and inspect a Request Packet.



Turn on intercept and go ontrack.

Then once you load into OnTrack there should be information that allows each section of ontrack to load before you then “forward” it to the server. As you can see below once i logged in i can see the packets being sent to the server and me, with all the information with auth token and cookies.



Remediation Advice

- Enforce HTTPS (TLS) to encrypt all network communication and prevent sniffing.
- Avoid sending long lived tokens in headers over unsecured connections.
- Move to only HTTPonly secure cookies to store tokens where java script cannot access them.
- Implement token expiration and refresh mechanisms.

References

OWASP Session Management Cheat Sheet
https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html

Wireshark Official Website (Download & Docs) https://www.wireshark.org/

Burp Suite Repeater Documentation

https://portswigger.net/burp/documentation/desktop/tools/repeater

Okta Developer Blog – Why You Should Always Use HTTPS

https://developer.okta.com/blog/2019/08/22/why-you-should-always-use-https

OWASP Cheat Sheet Series Main Page (Optional for extra references)
https://cheatsheetseries.owasp.org/

Contact Details
Filipe Oliveira s222478779@deakin.edu.au

Pentest Leader Feedback.
Good work!