

CYTHON 2024

AUTHENTICATION SECURITY

PROBLEM STATEMENT: Development of Customizable browser extension in Email & Web Login portals.

TEAM NAME: SCRAMBLERS

MEMBERS:

M.JASWANTH KUMAR
PONNADA SAI SURYA
KOMAL KRISHNA
SAKETH MALLOJJALA

Browser Extension

GETTING SESSION PARAMETERS WITH LEAST NUMBER OF WARNINGS

The extension utilizes the **chrome.cookies** API to extract all the cookies with parameters by using a background script call **chrome.cookies.getAll()**.

Background Script:

When the background script calls **chrome.cookies.getAll()**, it's considered an operation within the scope of the granted permissions, and so no additional prompts will be necessary.

chrome.tabs API :

The **chrome.tabs** API in a Chrome extension allows interaction with the browser's tab system. This permission allows temporary access to the current active tab without triggering any warnings for the user.

document.cookie Property:

The **document.cookie** property is part of the DOM API and is used to read HTTP cookies associated with the current document. It returns a string containing all cookies associated with the current document in the form of key-value pairs.

Sending Extracted Cookies:

After extracting the cookies, you can use **chrome.runtime.sendMessage()** to send a message containing the cookies to the background script.

SENDING SESSION PARAMETERS TO A SPECIFIC SERVER

Creating JSON File:

Once the cookies are extracted, the extension creates a **.json** file which sends the data associated with cookies to a specified server API.

HTTP POST Request:

The extracted data is packaged into a request payload and sent via HTTP POST request to the defined server API endpoint using **axios**.

Server API Listening:

This server API listens for incoming POST requests on a specific route, typically extracted cookies within a specific interval of time.

DISABLING AUTO REFRESH FEATURE IN A HYPERLINK

Meta Tag:

To disable Auto-Refresh, we executed a **<meta>** tag in the text input field of the console tab.

LOCATING A HYPERLINK AND MODIFYING IT WITH A USER-SPECIFIED URL (TEST.COM)

parse_link Function:

The **parse_link()** function is responsible for reading configuration options from the Chrome extension options page using **chrome.storage.sync.get** API. It retrieves the values of **oldHref** and **newHref** from Chrome storage. Then, it calls the **replace_links()** with the retrieved values.

replace_links Function:

This function is intended to replace all occurrences of a specific URL (**oldHref**) with another URL (**newHref**) in the href attributes of anchor elements (<a> tags) on the current webpage. It first checks if both **oldHref** and **newHref** are provided. If either of them is missing, it returns false. It fetches all anchor elements on the page using **document.getElementsByTagName('a')**. Then, it iterates through each anchor element and replaces the href attribute value using the replace method.