Introduction

- DOM Cracker is a node.js framework which can be used for accessing the DOM of webpages.
- It act as a server.
- We may use it for providing customer various services.
- It act as a gateway between web server and client .
- It fetches the data(as DOM) from server, injects some scripts and send it to the customer/client .

DOM Cracker

It can be used for:-

- Content Filtering
- Proxy Chaining
- Stopping phishing activities
- Test Automation of web applications
- Web content analysis like task

and many other applications where we need to control the DOM.

Can make simple websites more responsive

- By making some rules like :-
  - Header is at the top , generally first direct child of body , and having tag.
  - Generally, second direct child of body is navigation bar having <ul> or <table> tag.
  - Last direct child of body is footer having tag.
  - Other all are body contents.

By making these rules we may cover most of the old websites and may make them more responsive

(or)

may remove most of the unwanted contents like <script>, advertisements etc. for speed browsing on slow network

(or)

may monitor real-time speed of client and may provide dynamic web content.

Say Good Bye to hackers ….
No more Phishing

- Most of the hackers make most visiting sites as their target .
- A solution and an approach
  - We may use this framework to get dom access and hence content .
  - We may extract out some most frequent texts and phrases ,
  - Use google search api for getting some top results ,
  - Match their contents with dom object contents ,
  - If same contents found then add it to phishing site list in database ,
  - Otherwise add to safe site list.

This is a learn from user visit approach to get most of the phishing sites on the web.

A way to product automation testing

- We may inject various event listeners like click , change , dblclick , etc. to track developer activities and record it in a script format.
- After any up gradation to the product , he may play previous scripts for different components of product to test it for any break/error.

Content filtering

- Today most of the services for content filtering is url based.
- This is done by some DNS like openDNS.
- But may be whole contents are not unsafe/unwanted for client .
- We may make certain rules :-
  - Like list of unsafe words/phrases.
  - Advertisement (or) <script>
  - may process images <img> to get unwanted one

  And may remove it from the DOM before sending to the client.