Abstract

JavaScript programs are always under the threat of being copied. Most browsers provide the way to access the code of JavaScript program so it is easily obtainable. Hence it is mandatory to protect the software. Watermarking and code obfuscation are the techniques used to safeguard the software. A Watermark cannot completely protect the code by getting stolen because a potential attacker can easily remove it. Code obfuscation cannot avoid code from being stolen; it only prevents others by understanding the logic of the program. A birthmark of the JavaScript program is the unique characteristics that it possesses. Heap Graph is used to depict the behaviour of a program as how it calls other objects so as to fulfil the desired functionality. It requires efficient merging of heap graphs generated at various points of time.
For that agglomerative clustering can be used. Frequent Subgraph Mining is used to find the subgraph that represents the unique behaviour of the program. At the end, the subgraph of genuine program is searched in the graph of the suspected program. Our aim is to survey about the system that can protect the JavaScript programs from being stolen.

References

Index Terms

Computer Science  Data Mining

Keywords

Dynamic Birthmark  Agglomerative Clustering  Frequent Subgraph Mining  Theft Identification