امنیت شبکه های کامپیوتری

فعالیت آزمایشگاهی

با توجه به ماهیت توزیع شده اجرای نرم افزار، نیاز به کنترل دسترسی به بخش های مختلف کد که در ماشین های مختلف اجرا می شود وجود دارد. یکی از این محیط های توزیع شده محیط وب است که صفحه های مختلف در بستر شبکه به هم دسترسی پیدا می کنند و این دسترسی در بسیاری اوقات موجب اجرای اپلیکیشن ها و سرویس های نهفته در آن می گردد. محیط توسعه نرم افزار NodeJS نیز مثال دیگری است که کد معمولا از بخش های مختلفی تشکیل می شود که نیاز به دسترسی روی شبکه به هم دارند.

برای امن سازی چنین بسترتوزیع شده ای، فناوری های متنوعی توسعه یافته است که یکی از آنها ابزار CORS است. با مراجعه به لینک زیر، در مورد نحوه عملکرد CORS مطالعه و فعالیت تعریف شده در آن را بر اساس دستور کار ارائه شده انجام و گزارش از آن تهیه کنید. با مطالعه بیشتر در مورد CORS انواع سیاست های کنترل دسترسی قابل تعریف در این محیط را در گزارش ذکر کنید.

What is CORS? Complete Tutorial on Cross-Origin Resource Sharing (auth0.com)

https://auth0.com/blog/cors-tutorial-a-guide-to-cross-origin-resource-sharing/

In this article, they take a look at CORS, the circumstances under which it is needed, the benefits it provides, and how to configure a Node + Express application to support CORS. You need to get the accompanying source code from GitHub.

Cross-Origin Resource Sharing (CORS) is a protocol that enables scripts running on a browser client to interact with resources from a different origin. This is useful because, thanks to the same-origin policy followed by `XMLHttpRequest` and `fetch`, JavaScript can only make calls to URLs that live on the same origin as the location where the script is running. For example, if a JavaScript app wishes to make an AJAX call to an API running on a different domain, it would be blocked from doing so thanks to the same-origin policy.