Prevenire gli attacchi ReDos in Node.js

Un attacco ReDos (Regular expression Denial of Service) sfrutta la complessità delle espressioni regolari usate per bloccare l'Event Loop di Node.js.

Nella sua forma più semplice, questo tipo di attacco (descritto in questa pagina), consiste nel passare richieste HTTP contenenti stringhe la cui lunghezza è pensata per costringere l'interprete JavaScript a valutare un'espressione regolare in un tempo che cresce notevolmente a seconda della lunghezza della stringa passata.

Nello specifico, se si utilizzano espressioni regolari nei percorsi tramite parametri o si effettuano validazioni su stringhe nei valori passati tramite GET o POST (tipicamente), si può essere esposti a questo rischio se i pattern usati:

1. Contengono raggruppamenti con ripetizione.
2. All'interno del gruppo ripetuto vi è un'ulteriore ripetizione e alternative.

Alcuni esempi:

```
(a-z+)+
([a-zA-Z]+)*
(ala?)+
([a-z0-9]+)+
```

In ExpressJS si può evitare di usare queste espressioni regolari nei path delle route sostituendole con parametri generici ove possibile e se si deve utilizzare un'espressione regolare la si usi con una lunghezza determinata. Ad esempio:

```
:lang([a-z]{2})?
```

In questo caso il parsing si arresta dopo il secondo carattere della stringa, quindi anche utilizzando stringhe complesse l’interprete non è costretto ad elaborarle nella loro interezza. Per i form, invece, si consiglia l’uso di un token nonce per evitare che il form venga inviato da remoto.