11. Introduction To Cryptographic Hashes - A Running Demo

We stored password directly in db. it is bad practice. We see news of user passwords leaked almost every day. many times it is about million of passwords that gets stolen. But problem is not as serious as it looks, because password is not stored as plain text in db. if user loses it’s ,lets say linkendin password, that does not mean that attackers can now login into many other websites where user has a account with same email and password.

Password in most cases, is stored in db by first applying cryptographic hash functions on them. These functions give reproducible results. for same input we always have same output. Its not an encryption function, it is something else. To understand what it is, lets try it out. For this we need to switch branch. Run this command-

**git checkout –b 2-password-storage origin/2-password-storage**

run  **git branch** to see in which branch you are.

Now run npm install to get all dependcies.

Now lets see use of hash function and we will discuss its properties and why we want to use it to process our password before saving it into database. Go to hash.js file in demos folder. For each of these demos you will find a task in package.json, that will allow you to run demo file. lets run hash file.

In hash.ts we take password monkey and we create hash of it. We print it on console. Lets run hash file-

npm run hash

we saw a long chnain of characters on console so this lomg chian of characters is one form of password hash. And it will be something like this and not direct result of password hash, that will be stored on db instead of original password. So in this if hackers hack the dbthey will have this hash and not original password. Later we will see that siply hashing is not enough.

Lets review our small program to unsderstamd what is going on.first we require crypto module. Go to its documentation.there we se that many cryptographic security parameters are avalible. Here we also have some primitives for hashing. For demo purpose we have ued those primitives, to hash our password.

var hash = crypto.createHash('sha256').update(password).digest('hex');

what we didi here was, we created a cryptographic hashing function using crypto module and we used hash named ‘sha256’, this is official name of special hashinh function.there are many hashing function. in our code we wnt use this function, this is for demo of how to use crypto module that comes with node, to quickly generate hash of password. The using digest we specify the output format that we want to generate.