



Professional Skills
Agile Fundamentals
Jira
Git
Databases Introduction
Java Beginner
Maven
Testing (Foundation)
Java Intermediate
HTML
CSS
Javascript
Spring Boot
Selenium
Sonarqube
Advanced Testing (Theory)
Cucumber
MongoDB
Express
NodeJS
React
Express-Testing
Networking
Security
<div><div></div>Security Basics</div>
<div><div></div>Authentication and Authorization</div>
<div><div></div>Security Attacks</div>
<div><div></div>Passwords</div>
<div><div></div>Hashing vs Encrypting vs Encoding</div>
<div><div></div>Introduction to DevSecOps</div>
<div><div></div>Pen Testing, DAST & SAST</div>

Hashing vs Encrypting vs Encoding

Contents

- [Overview](#)
- [Encoding](#)
- [Encrypting](#)
- [Hashing](#)
- [Tutorial](#)
- [Exercises](#)

Overview

In this module we are going to cover the differences between Hashing, Encrypting and Encoding. We will also look at use cases for each one.

Encoding

Encoding is the process of converting data from one form to another.

You would do this for many reasons in the world of computers. For example, media files are often encoded to save disk space.

But the conversion can be reversed and anyone can reverse the process.

Encrypting

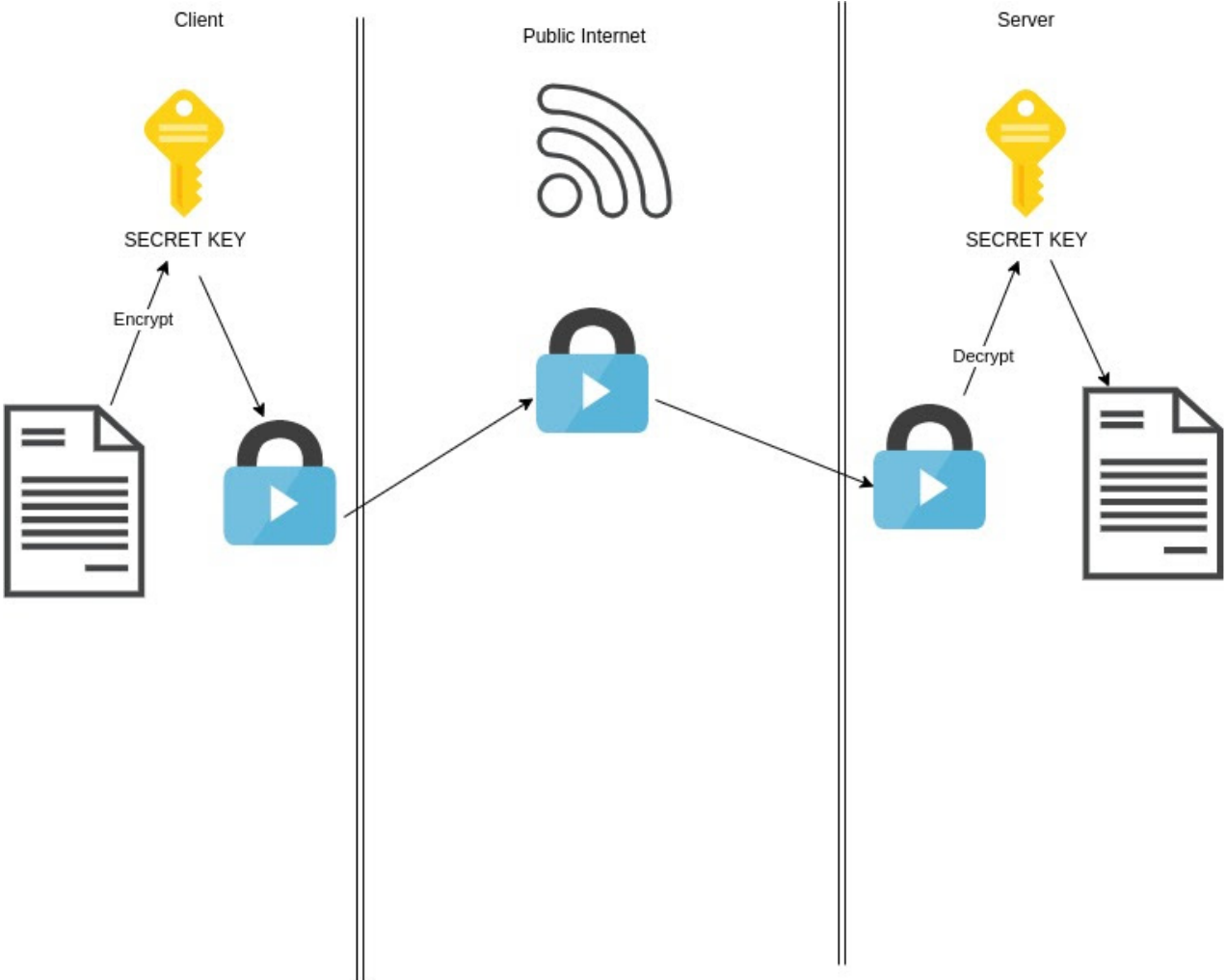
Encrypting is similar to encoding but the main difference is you convert data into a different form to keep it **secret**.

You would perform the conversion using a key which is kept private, normally called a **secret key**.

Encryptions can be reversed or "decrypted" but you can only reverse the process if you have the secret key.

During web development, if you want to send data from the client to the server. You will have to encrypt the data to stop unauthorised users from gaining access to that data during transit.

○ Introduction to OWASP
Cloud Fundamentals
AWS Foundations
AWS Intermediate
Linux
DevOps
Jenkins Introduction
Jenkins Pipeline
Markdown
IDE Cheatsheet



Hashing

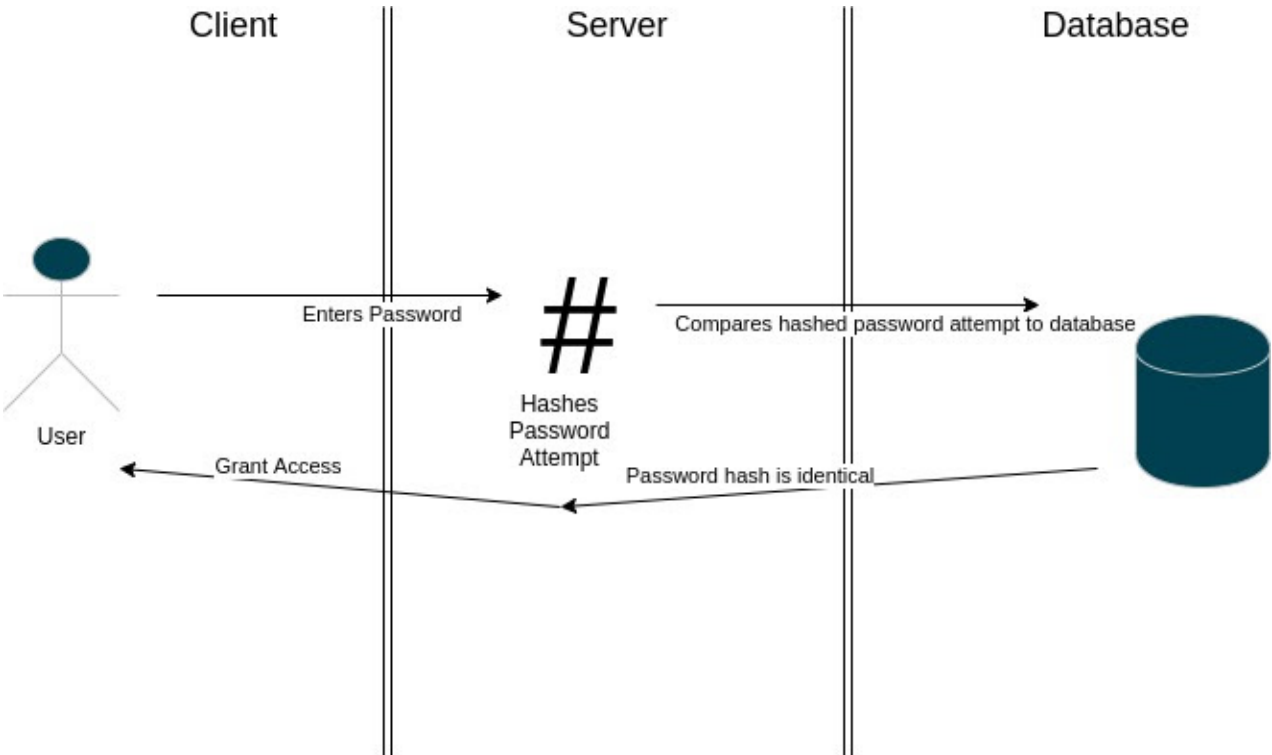
Hashing is when a computer takes a number of characters and chops them up and mixes them up. So it has a very similar output to encoding and encrypting but the main difference is that it **cannot be reversed**.

A good use case for hashing is storing passwords. If you store hashed passwords in a database this stops unauthorised users gaining access to accounts.

User	Password
ben	fcoijwei05394ifc£\$%

But then, the question is, how do you check if a password is correct if the stored version is hashed? And the answer is, if a user tries to log in with a password, you hash their attempt, and if it returns the same value as the hashed value stored then the password is correct.

This diagram will illustrate the process.



Tutorial

There is no tutorial for this module.

Exercises

There are no exercises for this module.