PHP: usare le funzioni password_hash() e password_verify()

Nell'era digitale moderna, proteggere i dati sensibili degli utenti è una priorità assoluta per ogni sviluppatore web. Una delle tecniche di sicurezza più utilizzate per proteggere le password degli utenti è l'hashing delle password. PHP, uno dei linguaggi di programmazione web più diffusi, offre due funzioni integrate per l'hashing delle password: password_hash() e password_verify().

La funzione `password_hash()` di PHP è utilizzata per creare l'hash di una password. Questa funzione prende in input la password in chiaro e restituisce una stringa contenente l'hash della password. L'hash generato è una stringa casuale e irripetibile, che viene utilizzata per verificare la correttezza della password in futuro.

La funzione `password_hash()` utilizza una tecnica di hashing sicura e avanzata. Questa tecnica utilizza un algoritmo di hashing a senso unico che rende molto difficile recuperare la password originale a partire dall'hash generato. Inoltre, la funzione `password_hash()` genera automaticamente un salt casuale, che viene utilizzato per rendere l'hash ancora più sicuro e difficile da violare.

La funzione `password_verify()` di PHP viene utilizzata per verificare se una password inserita dall'utente è corretta o meno. Questa funzione prende in input la password in chiaro e l'hash generato dalla funzione `password_hash()`, e restituisce un valore booleano che indica se la password è corretta o meno.

La funzione `password_verify()` utilizza l'hash della password per generare un nuovo hash della password inserita dall'utente e confrontare i

due hash. Se i due hash corrispondono, significa che la password inserita dall'utente è corretta. In caso contrario, la funzione restituisce false.

In generale, l'utilizzo delle funzioni `password_hash()` e `password_verify()` di PHP è molto semplice e intuitivo. Tuttavia, è importante ricordare che queste funzioni devono essere utilizzate correttamente per garantire la massima sicurezza delle password degli utenti.

In primo luogo, è importante utilizzare la funzione `password_hash()` per creare l'hash iniziale della password all'atto della creazione di un nuovo profilo utente, come durante la registrazione.

In secondo luogo, è importante utilizzare la funzione `password_verify()` per verificare la correttezza della password inserita dall'utente solo quando è necessario, ad esempio durante il login dell'utente. Inoltre, è importante utilizzare questa funzione correttamente, passando come parametro l'hash generato dalla funzione `password_hash()` e non la password in chiaro.

Ecco un esempio di come utilizzare le funzioni `password_hash()` e `password_verify()` di PHP:

```
// Creazione dell'hash della password
$password_in_chiaro = 'password_segreta';
$hash_password = password_hash($password_in_chiaro,
PASSWORD_DEFAULT);

// Memorizzazione dell'hash della password in un
database o file
// ...

// Verifica della correttezza della password inserita
dall'utente

$password_utente = trim($_POST['password']);
```

```
if (password_verify($password_utente,
$hash_password)) {
    // La password inserita dall'utente è corretta
} else {
    // La password inserita dall'utente non è
corretta
}
```

In questo esempio, prima viene creata l'hash della password utilizzando la funzione `password_hash()`, passando come parametro la password in chiaro e il parametro `PASSWORD_DEFAULT` per indicare l'utilizzo dell'algoritmo di hashing Bcrypt.

Successivamente, l'hash della password viene memorizzato in un database o file per l'utilizzo futuro.

Infine, viene verificata la correttezza della password inserita dall'utente utilizzando la funzione `password_verify()`.