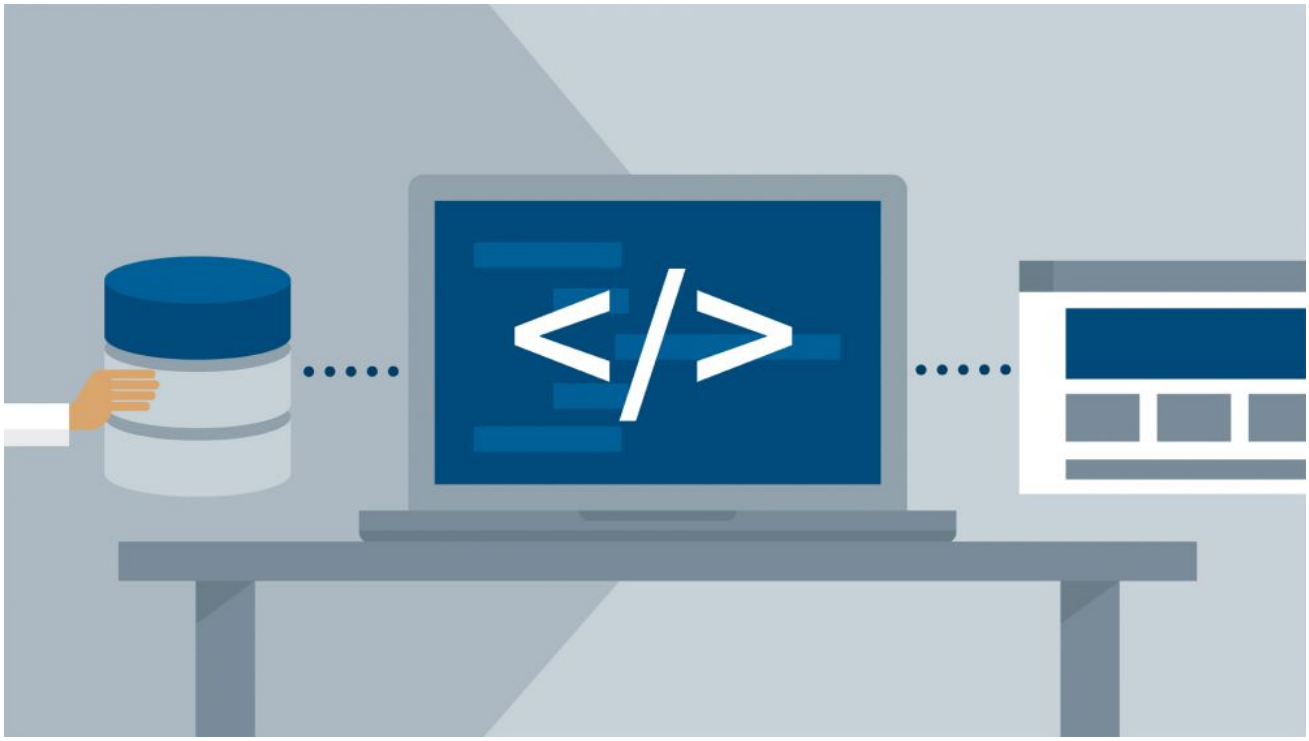
Mengamankan Ajax Request pada PHP

Oleh Adi Sumaryadi



Ajax seringkali digunakan oleh banyak web programmer untuk mendapatkan data tertentu tanpa me-load keseluruhan halaman. Namun terkadang banyak celah yang dapat digunakan oleh para peretas dari pengunaan ajax ini.

Ajax seringkali digunakan oleh banyak web programmer untuk mendapatkan data tertentu tanpa me-load keseluruhan halaman. Namun terkadang banyak celah yang dapat digunakan oleh para peretas dari pengunaan ajax ini. Oleh karenanya perlu adanya beberapa optimasi dari sisi keamanan. Beberapa hal yang dapat dilakukan adalah:

1. Membatasi Akses dari host yang tidak dikenal.
Cara ini untuk menghindari Cross Site Scripting yaitu dengan melakukan validasi pada bagian atas sebelum pengembalian data dilakukan berdasarkan parameter yang direquest.

```
header('Content-Type: application/json');

if (isset($_SERVER['HTTP_ORIGIN'])) {
    $address = 'http://' . $_SERVER['SERVER_NAME'];
    if (strpos($address, $_SERVER['HTTP_ORIGIN']) !== 0) {
        exit(json_encode([
            'error' => 'Invalid Origin header: ' . $_SERVER['HTTP_ORIGIN']
        ]));
```

```
    }
} else {
    exit(json_encode(['error' => 'No Origin header']));
}
```

2. Mengirimkan Token Keamanan
Metode kedua ini bisa juga dilakukan yaitu dengan cara menyamakan token yang dikirim dengan token yang ada di script ajax. Caranya adalah sebagai berikut:

1. Generate token :

```
session_start();
if (empty($_SESSION['csrf_token'])) {
    $_SESSION['csrf_token'] = bin2hex(random_bytes(32));
}
```

2. Tambahkan token yang telah di generate via meta :

```
<meta name="csrf-token" content="<?= $_SESSION['csrf_token'] ?>">
```

3. Lakukan Pemanggilan Ajax dengan mengirimkan Meta seperti dibawah ini :

```
$.ajaxSetup({
    headers : {
        'CsrfToken': $('meta[name="csrf-token"]').attr('content')
    }
});
```

4. Lakukan pengecekan di Script Ajax anda :

```
session_start();
if (empty($_SESSION['csrf_token'])) {
    $_SESSION['csrf_token'] = bin2hex(random_bytes(32));
}

header('Content-Type: application/json');

$headers = apache_request_headers();
if (isset($headers['CsrfToken'])) {
    if ($headers['CsrfToken'] !== $_SESSION['csrf_token']) {
        exit(json_encode(['error' => 'Wrong CSRF token.']));
    }
} else {
    exit(json_encode(['error' => 'No CSRF token.']));
```

```
}
```

3. Pengecekan User Login
Cara ketiga ini dikhususkan untuk ajax yang berjalan pada lingkungan yang telah login, jadi saat ajax script di panggil pastikan harus dicek terlebih dahulu sudah login atau belum.

Semoga bermanfaat.

Kata Kunci : PHP Ajax