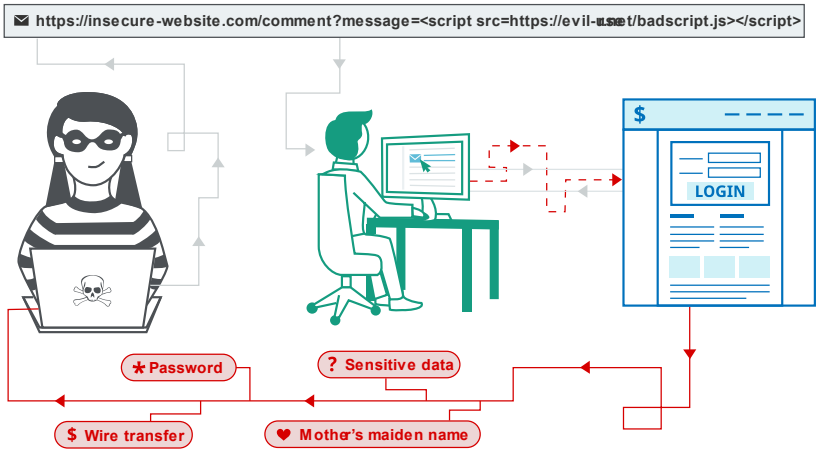
Cross-Site Scripting (XSS)

Apa itu ‘XSS’?

Cross-site scripting (atau lebih dikenali sebagai XSS) adalah sebuah kelemahan sekuriti web yang memberikan ‘attacker’ untuk mendominasi interaksi diantara ‘user’ yang mempunyai aplikasi yang lemah. Ia member ‘attacker’ untuk memintas ‘same-origin policy’, yang direka khas untuk mengasingkan kepelbagaian website dari satu sama lain. Kelemahan cross-site scripting selalunya terjadi apabila attacker menyamar menjadi mangsa, untuk melaksanakan tingkah laku yang mampu memberi kesan dalam mencapai segala akses data user. Jika mangsa mempunyai ‘privileged access’ dalam sesebuah aplikasi, maka ‘attacker’ mungkin dapat untuk mendapat ‘full control’ dalam semua fungsi dan data aplikasi tersebut.



Jenis-jenis serangan ‘XSS’?

- **Reflected XSS**, merupakan skrip yang direka berniat jahat dari ‘HTTP request’ semasa.
- **Stored XSS**, merupakan skrip yang direka berniat jahat dari pangkalan data website.
- **DOM-based XSS**, merupakan kelemahan yang wujud dalam ‘client-side code’ bukannya ‘server-side code’.

Reflected XSS

Reflected XSS merupakan variasi paling ringkas dalam cross-site scripting. Ia muncul apabila sesebuah aplikasi menerima data dalam bentuk ‘HTTP request’ dan memasuki data dalam tindak balas perantaraan dengan cara yang tidak selamat.

Disini sebuah contoh yang ringkas mengenai kelemahan reflected XSS:

https://insecure-website.com/status?message=Pokok+Rambutan

* Status: Pokok Rambutan *

Aplikasi ini tidak melaksanakan apa-apa pemprosesan data, maka ‘attacker’ mudah untuk membina sebuah ‘attack seperti ini:

https://insecure-website.com/status?message=<script>/+Pokok+Manggis+Disini+*/</script>*

* Status: <script>/* Pokok Manggis Disini */</script> *

Jika mangsa melawat URL yang dibina oleh ‘attacker’ itu maka skrip yang dibina oleh ‘attacker’ dapat dilaksanakan dalam browser mangsa. Dalam konteks dimana mangsa tidak selamat dalam sesi di aplikasi tersebut. Disaat itu, skrip itu mampu melaksanakan apa-apa tindakan, mencuri segala data, dimana mangsa mempunyai akses.

Stored XSS

Stored XSS (atau lebih dikenali sebagai ‘persistent or second-order XSS’) timbul apabila sesebuah aplikasi menerima data daripada sumber yang tidak dikenali dan termasuk data yang kemudian muncul dalam HTTP responses dalam keaadan yang kurang selamat.

Data berkenaan mungkin dikemukakan kepada aplikasi HTTP requests; sebagai contoh, komen dalam sesebuah post blog, user nicknames dalam ruang chat, atau butiran perhubungan order pelanggan. Dalam kes yang lain, data juga mungkin akan sampai melalui sumber yang tidak dipercayai; sebagai contoh, aplikasi webmail yang memaparkan mesej yang diterima melalui SMTP, sebuah aplikasi marketing memaparkan siaran media social atau aplikasi pemantauan rangkaian yang memaparkan ‘packet data melalui trafik rangkaian.

Ini merupakan sebuah contoh yang ringkas mengenai stored XSS vulnerability. Aplikasi papan mesej yang memberi ‘users’ untuk menghantar mesej, dimana akan dipaparkan kepada ‘users’ yang lain:

* Hai, ini message saya! *

Aplikasi ini tidak melaksanakan apa-apa pemprosesan data, maka ‘attacker’ mudah menghantar mesej yang mampu ‘attack’ ‘users’ yang lain:

* <script>/* Ini mesej jahat nyehehe… */</script> *

DOM-based XSS

DOM-based XSS (atau dikenali sebagai DOM XSS) muncul apabila aplikasi mempunyai client-side JavaScript yang data diproses daripada sumber yang tidak dikenali dalam keadaan yang tidak selamat, selalunya data ditulis semula kepada DOM.

Dalam contoh yang berikut, sebuah aplikasi menggunakan JavaScript untuk membaca value daripada sebuah ‘input field’ dan menulis semula kedalam elemen dalam HTML:

var search = document.getElementById('search').value;

var results = document.getElementById('results');

results.innerHTML = 'You searched for: ' + search;

Jika ‘attacker’ dapat mengawal nilai dalam ‘input field’, mereka dengan mudah mampu membina nilai yang jahat yang mampu membuatkan scrip ‘attacker’ melakukan:

You searched for: <img src=1 onerror='/ Perkara jahat disini... */'>*

Dalam kes yang biasa, ‘input field’ mungkin akan menjadi sebahagian daripada HTTP request, contohnya sebagai ‘URL query string parameter’, memberi peluang kepada ‘attacker’ untuk menyampaikan ‘attack’ dalam bentuk URL yang jahat, sama seperti keaadan yang di ‘reflected XSS’.