Procedia of Engineering and Life Science Vol. 2. No. 2 June 2022
Seminar Nasional & Call Paper Fakultas Sains dan Teknologi (SENASAINS 4th)
Universitas Muhammadiyah Sidoarjo

Techniques For Testing Website Security Using The Escaping Metacharacter Method

Teknik Menguji Keamanan Website Dengan Menggunakan Metode Escaping Metacharacter

Aditya Wira Utama, Arif Senja Fitrani
{wiraaditya0@gmail.com, asfjim@umsida.ac.id}

Program Studi Informatika, Fakultas Sains dan Teknologi, Universitas Muhammadiyah Sidoarjo

Abstract*. Information has become a very important need and has even become a basic need in this day and age. Much of the information available is confidential and not everyone has access to that information. CrossSite Scripting is a type of injection attack against a site by relying on weaknesses from the website or the user itself. Attackers try to steal data, confidentiality and website structure with certain commands through code scripting which is very contrary to the principles of information system security. So that the basic principles of information system security can be fulfilled, it is necessary to conduct research with the aim of finding loopholes and the effect of certain treatments for CrossSite Scripting attacks on websites under controlled conditions and closing the vulnerability gaps of the website. The research was conducted using the escaping metacharacter method which functions to convert special characters into HTML format. This method also functions as a filter on input so that the script that is injected by the user will not be executed by the browser and will be considered as normal input. From the results of the study, it shows that by entering certain characters or words as a rule (filter) which is the hallmark of the xss script, a pattern will be obtained which can later be used as a determinant of whether the input is really an ordinary string or script injection. The research contribution to be achieved is the importance of doing penetration testing on an information system for Agencies, Institutions and Companies, to always be able to recognize, analyze and respond to attacks on information systems and provide security and comfort to users of the information system.*

Keywords – *Cross Site Scripting; Information; Metacharacter; Vulnerabilities*

Abstrak*. Informasi menjadi kebutuhan yang sangat penting dan bahkan menjadi kebutuhan pokok di zaman ini. Informasi yang tersedia banyak yang bersifat rahasia dan tidak semua orang mendapatkan akses untuk informasi tersebut. CrossSite Scripting merupakan jenis serangan injection terhadap situs dengan mengandalkan kelemahan dari website atau pengguna itu sendiri. Penyerang berusaha untuk mencuri data, kerahasiaan dan struktur website dengan perintah tertentu melalui code scripting yang dimana hal itu sangat bertolak belakang dengan prinsip keamanan sistem informasi. Agar prinsip dasar keamanan sistem informasi dapat terpenuhi, maka perlu dilakukan penelitian dengan tujuan untuk mencari celah dan pengaruh perlakuan tertentu serangan CrossSite Scripting pada website dalam kondisi yang terkendalikan serta menutup celah kerentanan dari website tersebut. Penelitian dilakukan dengan metode escaping metacharacter yang dimana berfungsi untuk merubah karakter-karakter khusus menjadi format HTML. Metode ini juga berfungsi sebagai filter pada inputan agar script yang diinject oleh user tidak akan dijalankan oleh browser dan akan dianggap sebagai inputan biasa. Dari hasil penelitian menunjukkan dengan memasukkan karakter atau kata – kata tertentu sebagai rule(pemfilter) yang menjadi ciri khas dari script xss maka akan didapatkan sebuah pola yang nantinya dapat digunakan sebagai penentu apakah inputan tersebut benar – benar string biasa atau kah script injection. Kontribusi penelitian yang ingin dicapai adalah pentingnya melakukan penetration testing pada suatu sistem informasi Instansi, Lembaga dan Perusahaan, untuk senantiasa dapat mengenali, menganalisa dan menanggapi serangan pada sistem informasi serta memberikan keamanan dan kenyamanan pada pengguna sistem informasi tersebut.*

Kata Kunci – *Crosssite Scripting; Informasi; Metacharacter; Kerentanan*

I. PENDAHULUAN

Informasi menjadi kebutuhan yang sangat penting dan bahkan menjadi kebutuhan pokok di zaman ini. Informasi juga merupakan suatu hal yang bersifat sensitif, dalam hal ini informasi bisa digunakan sebagai hal baik maupun buruk[1]. Berdasarkan data yang dibuat oleh symantec, Indonesia berada di peringkat lima dunia negara yang paling banyak terserang pada tahun 2018 sebanyak 2,23% [2]. Hal ini juga di paparkan Badan Siber & Sandi Negara (BSSN) yang mencatat adanya 741 juta serangan di ruang internet Indonesia pada Januari - Juli 2021 [3]. Serangan XSS sendiri

adalah teknik yang menyisipkan suatu script ke situs web guna mendapatkan cookies yang mengarahkan korban ke malicious site yang mengandung malware atau phising, sehingga peretas bisa masuk ke dalam system dengan paksa dan juga mendapatkan data yang diinginkan. Serangan ini biasa dilakukan pada inputan atau URL (Uniform Resource Locator) yang ada dalam sistem [4].

Dengan menggunakan penetration testing kita bisa mendeteksi apakah injeksi yang dilakukan oleh peretas termasuk kategori berbahaya atau hanya sekedar mengidentifikasi aplikasi kita [5]. Dengan mengumpulkan semua informasi tentang system target, pencarian celah keamanan bisa dilakukan dengan manual atau secara otomatis. Selanjutnya setelah kita menemukan celah keamanan, kita menentukan target yang menjadi celah kerentanan, pemilihan tools dan exploit (kelemahan atau vulnerability sistem) yang tepat [6].

Langkah terakhir berupa patching bug (perbaikan kerentanan sistem) menggunakan metode escaping metacharacter untuk menghindari nilai apapun yang akan mengubah perilaku input [7]. fungsi metode escaping metacharacter pada kolom URL yang telah menangkap inputan dari form pencarian. Apabila request dari user mengandung script injection, maka nantinya script tersebut difilter terlebih dahulu menggunakan function htmlspecialchars() dan sistem pun tidak menjalankan script yang diinjekkan [8].

II. METODE

Data yang diambil dalam pengujian ini adalah data warga Desa Betro, Kecamatan Sedati, Sidoarjo yang berjumlah 3 jenis yang masing – masing telah dipersiapkan untuk dilakukan pengujian yaitu data warga, data mutasi warga dan data kepala keluarga. Pengujian akan dilakukan guna membangun keamanan website melalui metode escaping metacharacter, dengan cara menerapkan sebuah serangan terhadap website melalui URL maupun inputan yang ada menggunakan serangan XSS [9][10]. Kemudian menganalisa apakah metode escaping metacharacter terbukti efektif untuk mengamankan website dari serangan XSS.

Adapun pengujian dan pencarian celah pada program yaitu :

A. Pengujian Penetration Testing

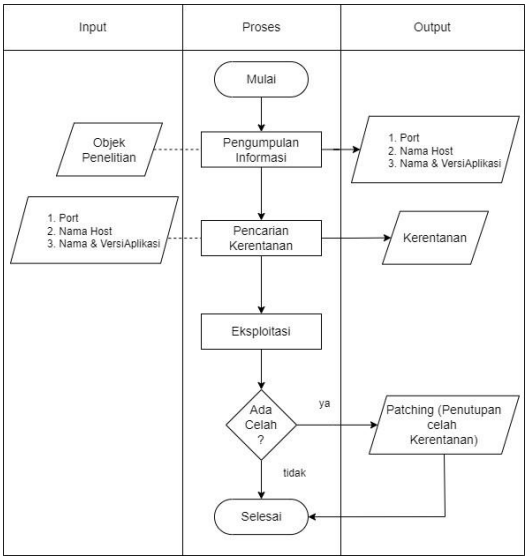
Pada tahapan ini peneliti terlebih dahulu akan mengumpulkan informasi pada program secara otomatis menggunakan fitur scanner pada tools yang digunakan untuk mengetahui kerentanan umum dan mencantumkannya dengan tingkat kerumitan eksploitasinya. Selanjutnya pada tahap eksploitasi, peneliti akan mencoba menginjekkan script ke dalam form pencarian yang nantinya di tangkap oleh form url apabila tidak menggunakan function escaping metacharacter.

B. Pengujian Fungi Escaping Metacharacter Terhadap Injection

Kemudian pengujian terakhir adalah pengujian fungsi metode escaping metacharacter pada kolom form url yang telah menangkap inputan dari form pencarian. Apabila request dari user mengandung script injection, maka nantinya script 30 tersebut difilter terlebih dahulu menggunakan function htmlspecialchars() dan sistem pun tidak menjalankan script yang diinjekkan

III. HASIL DAN PEMBAHASAN

Langkah awal pada pengujian ini adalah melakukan penetration testing (percobaan serangan) pada website yang akan dijadikan target. Langkah selanjutnya adalah melakukan pengamanan pada file index.php dan data-index.php terlebih dahulu dengan cara memberikan filtering berupa htmlspecialchars() pada file yang terdapat form inputannya [11][12]. Hal ini digunakan untuk memfilter setiap inputan yang dimasukkan oleh user [13][14]. Kemudian fungsi htmlspecialchars() diberikan pada parameter inputan tersebut adalah untuk menghindari serangan xss yang berasal dari url yang menggunakan method $_GET seperti pada script data-index.php. Berikut ini merupakan penggambaran alur dari penetration testing yang nantinya akan digunakan pada penelitian ini [15] :



Gambar 1. Alur penetration testing

A. Tahap Pengumpulan Informasi

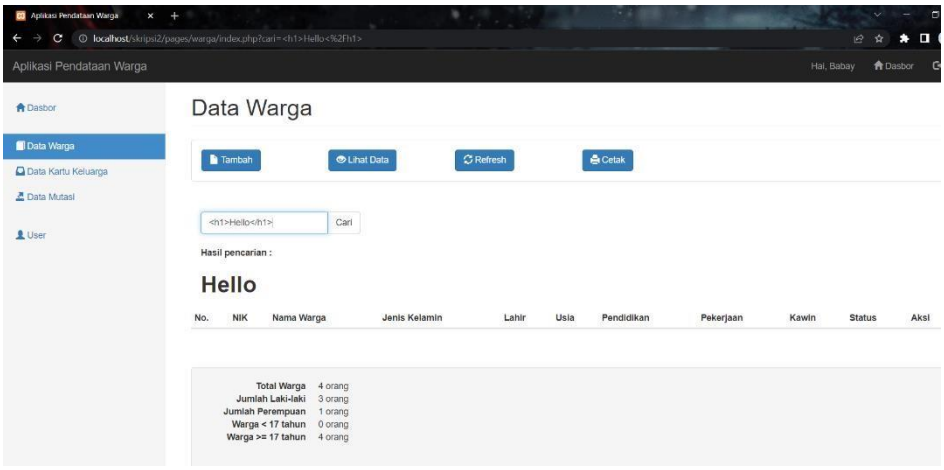
Dalam Tahap ini peneliti akan menggunakan sebuah tool yang Bernama NMAP yang berfungsi untuk melihat nama host yang tersedia dalam jaringan. Selain itu adanya IP raw juga untuk mengetahui layanan yang diberikan didalamnya dimana memuat nama dan juga versi aplikasi, sistem operasi lengkap dengan versinya. Informasi yang didapatkan dari tahap scanning ini adalah port yang terbuka, webserver yang digunakan beserta nomor versinya, bahasa pemrograman yang digunakan beserta nomor versinya.

```
└─$ nmap -v -A 127.0.0.1
Starting Nmap 7.92 ( https://nmap.org ) at 2022-04-04 20:55 WIB
NSE: Loaded 155 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 20:55
Completed NSE at 20:55, 0.00s elapsed
Initiating NSE at 20:55
Completed NSE at 20:55, 0.00s elapsed
Initiating NSE at 20:55
Completed NSE at 20:55, 0.00s elapsed
Initiating Ping Scan at 20:55
Scanning 127.0.0.1 [2 ports]
Completed Ping Scan at 20:55, 0.00s elapsed (1 total hosts)
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Initiating Connect Scan at 20:55
Scanning localhost (127.0.0.1) [1000 ports]
Discovered open port 21/tcp on 127.0.0.1
Discovered open port 3306/tcp on 127.0.0.1
Discovered open port 80/tcp on 127.0.0.1
Discovered open port 443/tcp on 127.0.0.1
Completed Connect Scan at 20:55, 0.01s elapsed (1000 total ports)
Initiating Service scan at 20:55
Scanning 4 services on localhost (127.0.0.1)
Completed Service scan at 20:55, 12.02s elapsed (4 services on 1 host)
NSE: Script scanning 127.0.0.1.
Initiating NSE at 20:55
Completed NSE at 20:55, 0.50s elapsed
Initiating NSE at 20:55
Completed NSE at 20:55, 1.29s elapsed
Initiating NSE at 20:55
Completed NSE at 20:55, 0.00s elapsed
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000035s latency).
Not shown: 996 closed tcp ports (conn-refused)
PORT     STATE SERVICE  VERSION
21/tcp   open  ftp      ProFTPD
80/tcp   open  http     Apache httpd 2.4.52 ((Unix) OpenSSL/1.1.1m PHP/7.4.27 mod_perl/2.0.11 Perl/v5.32.1)
| http-title: Welcome to XAMPP
|_Requested resource was http://localhost/dashboard/
|_http-favicon: Unknown favicon MD5: 56F7C04657931F2D0B79371B2D6E9820
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.52 (Unix) OpenSSL/1.1.1m PHP/7.4.27 mod_perl/2.0.11 Perl/v5.32.1
443/tcp  open  ssl/http Apache httpd 2.4.52 ((Unix) OpenSSL/1.1.1m PHP/7.4.27 mod_perl/2.0.11 Perl/v5.32.1)
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-favicon: Unknown favicon MD5: 6EB4A43CB64C97F76562AF703893C8FD
| http-title: Welcome to XAMPP
|_Requested resource was https://localhost/dashboard/
| tls-alpn:
|_  http/1.1
| ssl-cert: Subject: commonName=localhost/organizationName=Apache Friends/stateOrProvinceName=Berlin/countryName=DE
| Issuer: commonName=localhost/organizationName=Apache Friends/stateOrProvinceName=Berlin/countryName=DE
| Public Key type: rsa
| Public Key bits: 1024
| Signature Algorithm: md5WithRSAEncryption
| Not valid before: 2004-10-01T09:10:30
| Not valid after:  2010-09-30T09:10:30
| MD5:   b181 18f6 1a4d cb51 df5e 189c 40dd 3280
|_SHA-1: c4c9 a1dc 528d 41ac 1988 f65d b62f 9ca9 22fb e711
|_ssl-date: TLS randomness does not represent time
```

Gambar 2. Tahap scanning pada website target

B. Pencarian Kerentanan

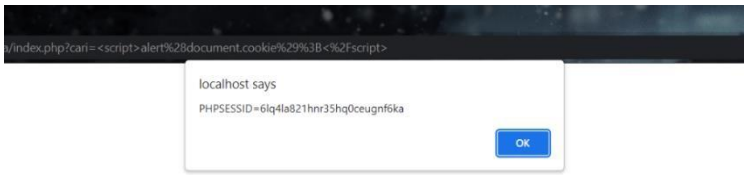
Tahap Selanjunya adalah tahap pencarian kerentanan xss secara manual yang terdapat pada halaman data warga. Pada halaman ini akan di uji pada form pencariannya apakah terdapat bug xss atau tidak. Langkah pertama akan di ujikan dengan menginputkan script html seperti <h1>Hello</h1> jika pada halaman pencarian menampilkan tulisan Hello dengan besar maka form pencarian tersebut bisa di rentan terhadap serangan xss. Seperti yang ditunjukkan pada gambar 2 form pencarian pada halaman Data Warga tersebut terdapat celah xss.



Gambar 3. Pencarian kerentanan pada halaman data warga

C. Tahap Eksploitasi

Tahap Selanjunya adalah tahap pencarian kerentanan xss secara manual yang terdapat pada halaman data warga. Pada halaman Langkah selanjutnya adalah melakukan pengujian terhadap script injeksi. Dimulai dari script xssinjection yang akan diuji dengan cara mencuri cookies dari admin baik melalui form pencarian ataupun url. Disini script yang digunakan untuk mencari cookies adalah <script>alert(document.cookies);</script>, seperti yang ditunjukkan pada gambar 3 di bawah ini, ketika script ini diinputkan, maka akan menampilkan cookies dari user yang sedang login pada website tersebut dan nantinya jika cookies user jatuh ke tangan yang tidak bertanggung jawab, maka peretas bisa menggunakan cookies tersebut untuk login tanpa harus menggunakan username dan password user.



Gambar 4. Pencurian cookies user pada halaman website

Seperti yang kita lihat pada gambar 4, script xss injection yang telah diinjekkan nantinya akan dijalankan pada website yang belum terfilter oleh fungsi escaping htmlspesialchar(). Akan tetapi pada studi kasus xss-reflected script ini hanya berlaku untuk satu kali injeksi, jika halaman di refresh kembali, maka halaman tersebut sudah kembali seperti semula dan seakan – akan tidak terjadi apapun. Hal inilah yang membuat serangan xss dianggap sepele oleh sebagian pihak. Namun serangan ini tetap berbahaya untuk sebuah website.

D. Tahap Patching Bug (Penutupan Celah)

Untuk menghindari user menginput script xss-injection kembali pada website, maka inputan – inputan akan diberi fungsi escaping metacharacter htmlspecialchar(), begitu juga dengan request $_GET pada url yang menangkap inputan. maka kegiatan injection akan otomatis terfilter (tidak dijalankan) dan tidak dianggap sebuah request pada website.

```
<form action="index.php" method="get" id="search-
form" class="navbar-form navbar-left">
    <div class="input-group">
    <input class="form-control" type="text"
name="cari" autocomplete="off" autofocus=""
placeholder="Pencarian">
      <span class="input-group-btn"
title="clear">&times;
      <button class="btn btn-default"
type="submit">Search</button>
      </span>
    </div>
    <?php
      if(isset($_GET['cari'])){
        $cari = htmlspecialchars($_GET['cari']);
        echo "<br></br><p><b>Hasil pencarian :
".$cari."</b></p>";
   }
   ?>
   </form>
```

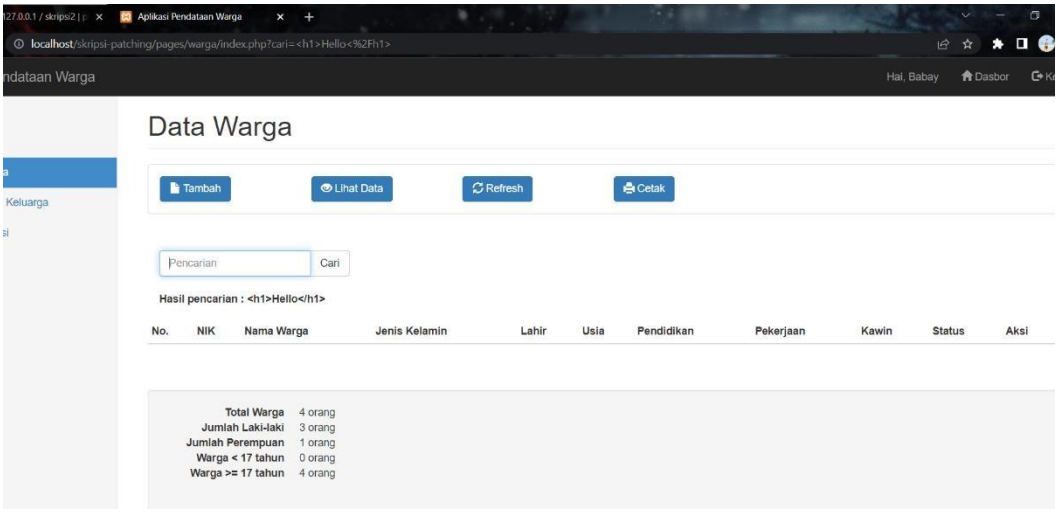
Gambar 5. Script Index.php

Pada file script index.php menjelaskan tentang form pencarian yang menggunakan method $_GET untuk menangkap data yang diinputkan dari user. Pada form pencarian, sangat rawan terjadinya serangan xss, maka dari itu kita akan mengamankan variable $cari yang akan menerima inputan user dan meneruskan request dari user menuju $_GET dari serangan xss-injection dengan cara memfilternya dengan fungsi escaping metacharacter htmlspecialchars().

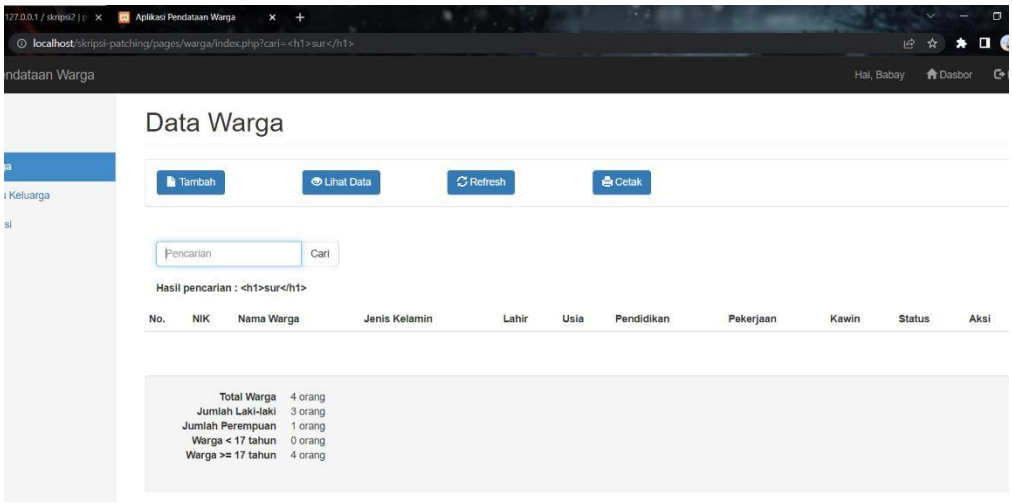
```
<?php
include('../../config/koneksi.php');
if (isset($_GET['cari'])) {
    $cari = htmlspecialchars($_GET['cari']);
 $query =  "SELECT *, TIMESTAMPDIFF(YEAR, `tanggal_lahir_warga`,
CURDATE()) AS usia_warga FROM warga where nik_warga like '%".$cari."%' OR
nama_warga LIKE '%".$cari."%' OR nama_warga LIKE '%".$cari."%' OR
tempat_lahir_warga LIKE '%".$cari."%' OR tanggal_lahir_warga LIKE '%".$cari."%' OR
jenis_kelamin_warga LIKE '%".$cari."%' OR alamat_ktp_warga LIKE '%".$cari."%' OR
alamat_warga LIKE '%".$cari."%' OR desa_kelurahan_warga LIKE '%".$cari."%' OR
kecamatan_warga LIKE '%".$cari."%' OR kabupaten_kota_warga LIKE '%".$cari."%'
OR provinsi_warga LIKE '%".$cari."%' OR negara_warga LIKE '%".$cari."%' OR
rt_warga LIKE '%".$cari."%' OR rw_warga LIKE '%".$cari."%' OR agama_warga LIKE
'%".$cari."%' OR pendidikan_terakhir_warga LIKE '%".$cari."%' OR pekerjaan_warga
LIKE '%".$cari."%' OR status_perkawinan_warga LIKE '%".$cari."%' OR status_warga
LIKE '%".$cari."%' ";
} else{
      $query = "SELECT *, TIMESTAMPDIFF(YEAR, `tanggal_lahir_warga`,
      CURDATE()) AS usia_warga FROM warga ";
}
$hasil = mysqli_query($db, $query);
$data_warga = array();
while ($row = mysqli_fetch_assoc($hasil)) {
$data_warga[] = $row;
}
```

Gambar 6. Script Data-Index.php

Pada file script data-index.php menjelaskan tentang bagaimana proses request data dari $cari yang menggunakan method $_GET diterima dan diproses ke database untuk menampilkan data yang diminta dari user sesuai apa yang diinputkan di kolom pencarian. Untuk menghindari serangan xss di injekkan lewat url, maka kita akan mengamankan $cari yang membawa sekaligus mengembalikan request dari dan menuju index.php dengan cara memfilternya dengan fungsi escaping metacharacter htmlspecialchars().



Gambar 5. Halaman data Warga setelah menerima serangan xss dari inputan



Gambar 6. Halaman data Warga setelah menerima serangan xss dari inputan

Pada gambar 5 menjelaskan bahwa ketka script xss dimasukkan lewat form input, maka fungsi escaping metacharacter *htmlspecialchars()* akan melakukan filtering pada inputan sehingga browser tidak akan menjalankan script xss yang diinputkan oleh user. Dan menganggap script yang dimasukkan menjadi inputan biasa.

Begitu juga halnya pada gambar 6, ketika user mencoba memasukkan script xss lewat form url diantara parameter request data user, maka fungsi escaping metacharacter *htmlspecialchars()* akan memfilter request $_GET agar bebas dari script xss yang diinputkan oleh user. Sehingga script yang diinputkan pada form url akan diubah menjadi request biasa yang diinputkan oleh user.

Dalam hal ini semua pengujian telah dilakukan mulai dari pengujian penetration testing hingga pengujian fungsi escaping metacharacter semua hasil uji berjalan sesuai yang diharapkan dan sesuai dengan yang dipaparkan dalam bab sebelumnya Seperti yang bisa dilihat dari script yang ada pada script file index.php dan data-index.php fungsi escaping metacharacter htmlspecialchars() berfungsi untuk merubah karakter – karater khusus menjadi format html. Dengan memasukkan karakter atau kata – kata tertentu sebagai rule(pemfilter) yang menjadi ciri khas dari script xss maka nantinya akan didapatkan sebuah pola yang nantinya dapat digunakan sebagai penentu apakah inputan tersebut benar – benar string biasa atau kah script injection.

IV. KESIMPULAN

Akhir dari pengujian dan analisa yang telah dilaksanakan pada bab sebelumnya dapat disimpulkan sebagai berikut: Dengan melakukan penetration testing pada suatu website, kita bisa mengetahui dan menentukan kelemahan dan serangan yang dapat terjadi terhadap kerentanan suatu sistem sedini mungkin dan dapat langsung diperbaiki sebelum terjadinya serangan terhadap sistem. Salah satu cara pencegahan serangan injection adalah dengan memfilter kata dan karakter yang masuk karena selalu ada celah untuk menyerang selama ada inputan user.

UCAPAN TERIMA KASIH

Peneliti menyampaikan terima kasih kepada Bapak Tejo Wibowo dan Ibu Suryawati selaku orang tua yang telah memberi doa dan motivasi selama ini. Bapak Evi Sukarijanto selaku Sekretaris Desa Betro yang turut andil untuk membantu mengizinkan penelitian pada website desa. Putra Angga Reksa rekan satu kelas yang senantiasa memberikan dukungan, canda tawa selama perkuliahan dan penyelesaian skripsi. Dan Ika Maulidiyah, M.Pd. saudara sepupu yang telah memberikan semangat, motivasi dan arahan selama proses penyelesaian skripsi.

REFERENSI

[1] B. A. Wisnu and N. J. K. P, "Prediksi Serangan Scripting Lintas Situs Menggunakan Mesin Belajar Algoritma," pp. 1–5, 2014.
[2] K. Devi, "Mendeteksi Aplikasi Web Berbasis XSS Kerentanan," 2017.
[3] Yulianingsih, "Melindungi Aplikasi dari Serangan CrossSite Scripting (XSS) Dengan Metode Metacharacter," 2017.
[4] S. Xss, M. Firefox, Y. T. Wahyuni, and A. M. Shiddiqi, "Rancang Bangun Add-ons Deteksi Cross Site," vol. 2, no. 1, pp. 1–4, 2013.
[5] R. Pangalila, A. Noertjahyana, and J. Andjarwirawan, "Penetration Testing Server Sistem Informasi Manajemen dan Website Universitas Kristen Petra."
[6] Fietyata Yudha. A. M, "Perancangan Aplikasi Pengujian Celah Keamanan Pada Aplikasi Berbasis Web," vol.2, 2018.
[7] H. Aliyasa Almaj Duddin and A. Senja Fitrani, "IMPLEMENTASI METODE IDS IPS DALAM PENGAMANAN FORM INPUT TERHADAP SERANGAN SQL-INJECTION Dan XSS," 2019.
[8] Herman Tolle, S.M, "Peningkatan Keamanan Web Terhadap Serangan Cross Site Scripting (XSS)," 2008.
[9] I Made Suartana, S. M, "Sistem Pengamanan Web Server Dengan Web Application Firewall (WAF)," vol.x, no.1, 2015, 42.
[10] Johanes Indra Homenta. R. E, "Menganalisa Sistem Keamanan Jaringan Berbasis Intrusion Prevention System dan Honeypot Sebagai Pendeteksi Dan Pencegah Malware (Studi Kasus PT. Karya Mitra Nugraha)," 3-4, 2015.
[11] Jhon Nicolas Siahaan, Y. M, "Aplikasi SMS Gateway Di PT. Mercaya Globe Sphere," 4-5, 2013.
[12] Gadhi Pranoto. R. D, "Rancang Bangun Aplikasi Terpadu," vol2, 2016.
[13] Dini Hariyati, R. A, "Pembangunan Sistem Informasi Rawat Jalan Berbasis Web Dengan Fitur Mobile Pada Puskesmas Tarok Kota Payakumbuh," 2017.
[14] Fauzan Masykur. F.P, "Aplikasi Rumah Pintar (*Smart Home*) Pengendali Peralatan Elektronik Rumah Tangga Berbasis Web," 95, 2016.
[15] Elisa Usada. Y. R, "Rancang Bangun Sistem Informasi," 43, 2012.