PENGUJIAN CELAH KEAMANAN INPUT VALIDATION PADA APLIKASI WEBSITE MENGGUNAKAN FRAMEWORK OWASP

Fefbi Septa Kristara[1], Mochamad Adhari Adiguna[2]

[1]Department Of Informatics, Universitas Pamulang, Banten, Indonesia, 15310
e-mail: [1]fefbisepta86@gmail.com, [2]dosen01864@unpam.ac.id

ABSTRACT

The importance of website security in the digital era is growing with the increasingly widespread use of the internet and technology. Website penetration testing is a security testing process carried out on a website with the aim of exploiting security gaps that may exist on the website. This can be done by testing for weaknesses in the website application, trying to carry out attacks on the website and application servers, and looking for security gaps in the network configuration. In this research the author will carry out penetration testing on input forms to test weaknesses in input validation on a website.aim of this research is to identify potential security vulnerabilities in the input form on the website, analyze the methods used to evaluate the system, and provide suggestions and references for improving the security of the website system.In this research the author uses the OWASP framework as a testing method. Owasp has methods and techniques for website testing which are packaged in the WSTG (Website Security Testing Guide) document. This document contains quite detailed methods and techniques for testing website applications.It is hoped that the results of this research can be a reference for website application developers and managers in order to improve the website security system that is being developed or managed. Apart from that, we hope that the results of this research can be a reference for cyber security practitioners in improving the techniques and methods used.

Keyword : Penetration, Exploit, Input Validation

ABSTRAK

Pentingnya keamanan website dalam era digital semakin berkembang dengan penggunaan internet dan teknologi yang semakin meluas. Pengujian penetrasi website adalah proses pengujian keamanan yang dilakukan pada website dengan tujuan mengeksploitasi celah keamanan yang mungkin ada pada website tersebut. Hal ini dapat dilakukan dengan cara menguji kelemahan pada aplikasi website, mencoba melakukan serangan pada pelayan website dan aplikasi, serta mencari celah keamanan pada konfigurasi jaringan. Pada penelitian ini penulis akan melakukan pengujian penetrasi pada formulir masukan untuk menguji kelemahan validasi masukan yang ada di dalam sebuah website. Tujuan dari penelitian ini adalah untuk mengidentifikasi potensi kerentanan keamanan pada formulir masukan di dalam website, menganalisis metode yang digunakan untuk mengevaluasi sistem, dan memberikan saran dan acuan dalam meningkatkan keamaman sistem website.Pada penelitian ini penulis menggunakan framework Owasp sebagai metode pengujian. Owasp memiliki metode dan teknik untuk pengujian website yang dikemas dalam dokumen WSTG(Website Security Testing Guide). Dokumen ini berisi metode dan teknik untuk melakukan penguian pada aplikasi website yang cukup detail.Hasil dari penelitian ini diharapkan bisa menjadi acuan untuk pengembang dan pengelola aplikasi website dalam rangka meningkatkan sistem keamanan website yang sedang dikembangan atau dikelola. Selain itu semoga hasil dari penelitian ini bisa menjadi referensi untuk praktisi kemanan siber dalam meningkatkan teknik dan metode yang digunakan.

Kata kunci : *Penetrasi, Eksploitasi, Input Validation*

1. PENDAHULUAN

Di masa sekarang ini, teknologi informasi dan internet berkembang dengan cepat. Website menjadi salah satu bentuk nyata dari perkembangan teknologi informasi dan internet. Website mempunyai peran penting dalam kehidupan masyarakat. Salah satu peran website adalah sebagai sarana umtuk berbagi dan mencari informasi. Perkembangan website sebagai sarana informasi, diiringi dengan berbagai macam ancaman siber.

Berdasarkan dari Pusat Operasi keamanan Siber Nasional (Pusopskamsinas) Badan Siber dan Sandi Negara (BSSN) mencatat 88.414.296 serangan siber telah terjadi sejak 1 Januari hingga 12 April 2020. Pada bulan Januari terpantau 25.224.811 serangan dan kemudian pada bulan Februari terekam 29.188.645 serangan lalu kemudian pada bulan Maret terjadi 26.423.989 serangan dan sampai dengan 12 April 2020 telah tercatat 7.576.851 serangan. Puncak jumlah serangan terjadi pada tanggal 12 Maret 2020 yang mencapai 3.344.470 serangan. (BSSN, 20-04-2020).

Salah satu serangan yang sering terjadi pada sebuah website adalah serangan pada formulir masukan dengan memanfaatkan kesalahan pada input validation dengan melakukan injeksi menggunakan kode atau skrip berbahaya. Berdasarkan data dari owasp, injection selalu masuk dalam 10 serangan yang sering terjadi.

Pengujian sistem keamanan menjadi salah satu faktor penting dalam menjaga keamanan website dari berbagai ancaman yang mungkin akan terjadi khususnya seraangan yang memanfaatkan kesalatan validasi pada form masukan. Pengujian sistem keamanan perlu dilakukan secara rutin, karena tidak ada jaminan bahwa suatu website akan aman dari serangan dalam jangka panjang.

Dari penjelasan diatas yang telah disampaikan maka, akan dilakukan penelitian untuk menguji sistem keamanan input validation yang ada pada aplikasi website dengan menggunakan framework owasp versi 4.1. Pengujian ini dilakukan untuk menguji bagaimana sebuah website mempunyai kerentanan pada form masukan, dan memberikan saran untuk perbaikan sistem keamanan jika ada sebuah kerentanan.

Tujuan dari penelitian ini adalah untuk mengetahui bagaimana serangan pada inpit validasi dilakukan. Dengan adanya penelitian ini diharapkan bisa menjadi referensi atau acuan untuk para pengembang aplikasi website untuk dalam rangka meningkatkan sistem keamanan pada aplikasi yang mereka kembangkan.

2. METODE

Pada penelitian ini pengujian dilakukan menggunakan teknik penetrasi atau penetration testing menggunakan framework OWASP yaitu WSTG (Website Security Testing Guide) versi 4.1. Pada penelitian ini menggunakan Damn Vulnerable Website Application (DVWA) sebagai target pengujian. Bidang yang diuji pada penlitian ini adalah Sql Injection, XSS Reflected, XSS Stored, File Inclusion, dan Command Injection.

Pada penelitian ini penguiji memasukan dan mengirim payload pada formulir masukan pada website dengan payload yang telah ditentukan berdasarkan WSTG. Payload yang dimaksud adalah sebagai berikut;

Tabel I. Data Payload

PAYLOAD XSS REFLECTED
<script>alert(‘test’)</script>
<Script>alert(‘test’)</Script>
%3Cscript%3Ealert%28%27test%27%29%3C%2Fscript%3E
<script>alert('test')</script>
<img src/onerror=alert(‘test)>
PAYLOAD XSS STORED
<script>alert(‘test’)</script>
<Script>alert(‘test’)</Script>
%3Cscript%3Ealert%28%27test%27%29%3C%2Fscript%3E
<script>alert('test')</script>
<img src/onerror=alert(‘test)>
PAYLOAD COMMAND INJECTION
ls -l
|ls -l
||ls -l
PAYLOAD FILE INCLUSION
/etc/passwd
/../../../etc/passwd
/etc/passwd%00
Php://filter/convert.base64-encode/Resource=/etc/passwd
data://text/plain;base64,PD9waHAgcGhwaW5mbygpOyA/Pg==
PAYLOAD SQL INJECTION

‘ union select 1,database()#
‘ Union Select 1,databses()#
‘ un/**/ion se/**/lect 1,database()#

3. HASIL

3.1 Hasil Pengujian XSS Reflected

Pada pengujian ini payload dimasukan satu per satu pada formulir input yang telah ditentukan yang ada pada website target. Berikut tampilan saat memasukan payload pada formulir input:

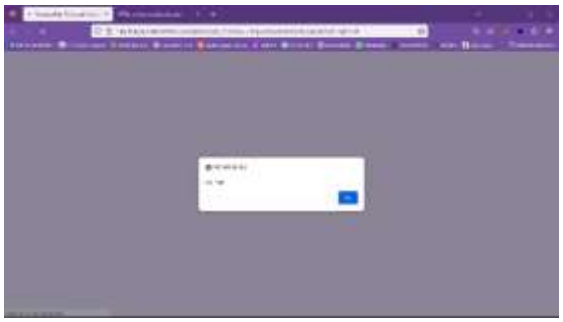
Gambar 1. Injeksi Payload XSS-Reflected

Setelah payload dimasukan dan dikirimkan akan didapatkan hasil pengujian. Berikut ini adalah hasil pengujian XSS-reflected. Hasil disajikan dalm table berikut:

Tabel II. Hasil Uji XSS reflected

Payload	Hasil
<script>alert(‘test’)</script>	TL
<Script>alert(‘test’)</Script>	TL
%3Cscript%3Ealert%28%27test %27%29%3C%2Fscript%3E	TL
<script>alert(' test')</script>	TL
<img src/onerror=alert(‘test)>	L

Dari table diatas terdapat keterangan hasil pengujian yang ditulis yaitu 4 TL (Tidak Lolos) dan 1 L (Lolos).



Gambar 2. Payload xss-reflected berhasil

Gambar diatas menunjukan bahwa payload yang di masukan berhasil lolos dari system keamaman website target. Dan payload telah dieksekusi oleh sistem.

3.2 Hasil Pengujian XSS Stored

Pada pengujian ini payload dimasukan satu per satu pada formulir input yang telah ditentukan yang ada pada website target. Berikut tampilan saat memasukan payload pada formulir input:

Gambar 3. Injeksi Payload XSS-Reflected

Setelah payload dimasukan dan dikirimkan akan didapatkan hasil pengujian. Berikut ini adalah hasil pengujian XSS-stored. Hasil disajikan dalm table berikut:

Tabel III. Hasil Uji XSS Stored

Payload	Hasil
<script>alert(‘test’)</script>	TL
<Script>alert(‘test’)</Script>	TL
%3Cscript%3Ealert%28%27test %27%29%3C%2Fscript%3E	TL
<script>alert(' test')</script>	TL
<img src/onerror=alert(‘test)>	L

Dari table diatas terdapat keterangan hasil pengujian yang ditulis yaitu 4 TL (Tidak Lolos) dan 1 L (Lolos).

Gambar 4. Payload xss-stored berhasil

Gambar diatas menunjukan bahwa payload yang di masukan berhasil lolos dari system keamaman website target. Dan payload telah dieksekusi oleh sistem.

3.3 Hasil Pengujian Command Injection

Pada pengujian ini payload dimasukan satu per satu pada formulir input yang telah ditentukan yang ada pada website target. Berikut tampilan saat memasukan payload pada formulir input:

Gambar 5. Injeksi Payload Command Injection

Setelah payload dimasukan dan dikirimkan akan didapatkan hasil pengujian. Berikut ini adalah hasil dari pengujian Command Injection. Hasil disajikan dalam table berikut:

Tabel IV. Hasil Uji Command Injection

Payload	Hasil
Ls -l	Tidak Lolos
|ls -l	Tidak Lolos
||ls -l	Tidak Lolos

Dari tabel diatas terdapat keterangan hasil pengujian yang ditulis yaitu Tidak Lolos.

3.4 Hasil Pengujian Local File Inclusion

Pada pengujian ini payload dimasukan satu per satu pada formulir input yang telah ditentukan yang ada pada website target. Berikut tampilan saat memasukan payload pada formulir input:

Gambar 6. Injeksi Payload Local File Inclusion

Setelah payload dimasukan dan dikirimkan akan didapatkan hasil pengujian. Berikut ini adalah hasil dari pengujian Local File Inclusion. Hasil disajikan dalam table berikut:

Tabel V. Hasil Uji Local File Inckusion

Payload	Hasil
/etc/passwd	TL
/../../../etc/passwd	TL
/etc/passwd%00	TL
Php://filter/convert.base64-encode/Resource=/etc/passwd	TL
data://text/plain;base64,PD9waHAgcGhwaW5mbygpOyA/Pg==	TL

Dari table diatas terdapat keterangan hasil pengujian yang ditulis yaitu Tidak Lolos.

3.5 Hasil Pengujian SQL Injection

Pada pengujian ini payload dimasukan satu per satu pada formulir input yang telah ditentukan yang ada pada website target. Berikut tampilan saat memasukan payload pada formulir input:

Gambar 7. Injeksi Payload SQL-Injection

Setelah payload dimasukan dan dikirimkan akan didapatkan hasil pengujian. berikut ini adalah hasil pengujian XSS-reflected. Hasil disajikan dalm table berikut:

Tabel VI. Hasil Uji SQL-Injectio

Payload	Hasil
‘ union select 1,database()#	L
‘ Union Select 1,databses()#	L
‘ un/**/ion se/**/lect 1,database()#	TL

Dari tabel diatas terdapat keterangan hasil pengujian yang ditulis yaitu 2 L (Lolos) dan 1 TL (Tidak Lolos).

Gambar 8. Payload sql injection berhasil

Gambar diatas menunjukan bahwa payload yang di masukan berhasil lolos dari system keamaman website target. Dan payload telah dieksekusi oleh sistem.

4. PEMBAHASAN

Pada pengujian penetrasi ini dilakukan menggunakan framework owasp berhasil menemukan beberapa celah keamana pada website target. Celah keamanan tersebut ialah celah yang memanfaatkan kesalahan sistem dalam memvalidasi masukan pada formulir masukan, sehingga dapat dimanfaatkan oleh orang tidak bertanggung jawab untuk melakukan tindak kejahatan yang dapat merugikan orang lain.

Pada pengujian XSS reflected terdapat 4 TL (Tidak Lolos) dan 1 L (Lolos). Dalam hal ini maksudnya adalah ada satu payload yang lolos dari sistem validasi yang terdapat pada formulir masukan tersebut. Payload yang lolos ini adalah *<img src/onerror=alert(‘test)>*. Payload ini dimodifikasi menggunakan tag html.

Pada pengujian command injection terdapat 4 TL (Tidak Lolos). Artinya dari 4 payloads yang dikirimkan tidak ada yang berhasil lolos dari sistem validasi. artinya website tersebut tidak ada kerentanan terhadap serangan command injection.

Pada pengujian local file inclusion terdapat 5 TL (Tidak Lolos). Artinya dari 5 payloads yang di kirimkan tidak ada yang berhasil lolos. Maka website tersebut tidak ada kerentaan terhadap serangan local file inclusion.

Pada pengujian sql injection terdapat 1 TL(Tidak Lolos) dan 2 L(Lolos). Artinya terdapat 2 payload yang lolos, yaitu *‘ union select 1,database()#* dan *‘ Union Select 1,databses()#.* Kode payload ini menggunakan kueri memanfaatkan kueri union dan memodifikasi case sensitive.

5. KESIMPULAN DAN SARAN

KESIMPULAN:

Dari hasil dan pembahasan penelitian ini, maka dapat diambil kesimpulan sebagai berikut:

1. Secara keseluruhan website tersebut memiliki kerentanan terhadap serangan cross site scripting dan sql injection.
2. Sistem validasi masukan pada website harus diperbaiki agar tidak terjadi serangan yang tidak diinginkan.
3. Metode dari owasp yang digunakan pada penelitian ini sangat efektif untuk pengujian kerentanan pada website.

SARAN:

Peneliti ini sudah dilakukan dengan semaksimal mungkin, namun peneliti menyadari masih banyak adanya keterbatasan dan kekurangan. Berdasarkan penelitian yang telah dilakukan maka saran yang di dapat diberikan adalah bagi peneliti selanjutnya diantara sebagai berikut;

1. Meningkatkan sistem keamanan website terutama pada kemampuan untuk memvalidasi masukan dari user agar dapat mencegah terjadinya serangan.
2. Melakukan pengujian kerentanan dan analisis secara berkala pada sistem secara menyeluruh.
3. Menerapkan metode OWASP untuk melakukan pengujian dan analisis untuk mengurangi resiko serangan.

6. UCAPAN TERIMA KASIH

Bapak Dr. Pranoto, selaku pemilik Yayasan Sasmita Jaya Grup dan pendiri Universitas Pamulang yang telah memberikan kesempatan kepada penulis untuk menimba ilmu di jenjang perkuliahan dengan mendirikan universitas yang terjangkau. Bapak Dr. E. Nurzaman A.M., Msi., M.M selaku Rektor Universitas Pamulang yang telah mengizinkan penulis untuk mengikuti dan menyelesaikan pendidikan Program Studi Teknik Informatika strata 1 di Universitas Pamulang. Bapak Achmad Udin Zailani, S.Kom., M.Kom, selaku Ketua Program Studi Teknik Informatika Universitas Pamulang atas dukungan dan pelajaran secara langsung maupun tidak langsung yang telah membantu penulis untuk

semangat menyelesaikan skripsi ini. Kedua orang tua penulis, yang telah memberikan dukungan lahir batin, moril, dan materil sehingga penulis dapat menyelesaikan pendidikan di Program Studi Teknik Informatika Universitas Pamulang. Bapak Mochamad Adhari AdigunaS.St., M.Kom, Selaku Dosen Pembimbing skripsi saya. Teman dan kerabat yang selalu mendukung dalam penelitian ini.

DAFTAR PUSTAKA

[1] K. Nagendran, A. Adithyan, R. Chethana, P. Camillus and B. S. V. K. Bala, "Web Application Penetration Testing," *International Journal of Innovative Technology and Exploring Engineering,* vol. 8, no. 10, pp. 1029-1035, 2019.

[2] M. Koprawi, "Dampak dan pencegahan serangan file inclusion: perspektif developer," *Jurnal Nasional Informatika dan Teknologi Jaringan,* vol. 4, no. 2, pp. 281-285, 2020.

[3] i. o. riandhanu, "analisis metode open web application security project (owasp) menggunakan penetration testing pada keamanan webiste absensi," *jurnal informasi dan teknologi,* vol. 4, no. 3, pp. 160-165, 2022.

[4] a. s. i gede sanjaya, a. gusti made sasmita and s. dewa made arsa, "evaluasi keamanan website lembaga x melalui penetrasion testing menggunakan framework isssaf," *jurnal ilmiah merpati,* vol. 8, no. 2, pp. 114-124, 2020.

[5] f. fachri, a. fadil and i. riadi, "analisis keamanan webserver menggunakan penetration test," *jurnal informatika,* vol. 8, no. 2, pp. 183-190, 2021.

[6] S. Hidayatulloh and D. Saptadiaji, "Penetration testing pada website universitas ars menggunakan open web application security project(owasp)," *Jurnal Algoritma,* vol. 19, no. 1, pp. 77-86, 2021.

[7] R. Y. Andriani, P. Hendradi and S. Nugroho, "Meningkatkan keamanan terhadap sql injection studi kasus sistem kepegawaian BNN," *Indonesian Journal of Business Intelligence,* vol. 6, no. 1, pp. 34-42, 2023.

[8] guntoro, l. costaner and musfawati, "analisis keamanan web server open journal system (ojs) menggunakan metode issaf dan owasp (studi kasus ojs universitas lancang kuning)," *jurnal ilmiah penelitian dan pembelajaran informatika,* vol. 5, no. 1, pp. 45-55, 2020.

[9] L. H. Yanti, I. and B. Cut, "Analisa keamanan web server dari serangan remote os command injection pada instansi pemerintahan kota banda aceh," *Jurnal Riset dan Inovasi Pendidikan,* vol. 1, no. 2, pp. 92-98, 2019.

[10] G. A. Septiawan, K. W. S. Irawan, I. Mayasari and I. M. E. Listartha, "Analisis kerentanan xss dan rate limiting pada website sman 8 denpasar menggunakan framework owasp zap," *Jurnal Informatika Upgris,* vol. 8, no. 1, pp. 99-101, 2022.

[11] A. I. Rafeli, H. B. Seta and I. W. Widi, "Pengujian celah keamanan menggunakan metode owasp web security testing guide (wstg) pada website xyz," *Jurnal Informatik,* vol. 18, no. 2, pp. 97-103, 2022.

[12] I. O. Laleb, "Analisis cross-site Scripting (xss) injection – reflected xss and stored xss mengggunakan framework owasp 10," *Jurnal Ilmiah Flash,* vol. 8, no. 1, pp. 36-42, 2022.

[13] E. Saad, J. Zold, J. B. Choi, J. Espunya, M. P. Tien, M. Clayton, R. Mitchell, R. Jain, T. Argoni and V. Drake, "www-project-web-security-testing-guide," 21 04 2020. [Online]. Available: https://owasp.org/www-project-web-security-testing-guide. [Accessed 5 7 2023].

[14] PortSwigger Ltd, "cross-site-scripting/cheat-sheet," portswigger ltd, 2023. [Online]. Available: https://portswigger.net/web-security/cross-site-scripting/cheat-sheet. [Accessed 5 7 2023].

[15] PortSwigger Ltd, "sql-injection/cheat-sheet," portswigger ltd, 2023. [Online]. Available: https://portswigger.net/web-security/sql-injection/cheat-sheet. [Accessed 5 7 2023].