Jurnal Sains Komputer & Informatika (J-SAKTI)
Volume 7 Nomor 2, September 2023, pp. 815-827
ISSN: 2548-9771/EISSN: 2549-7200
https://tunasbangsa.ac.id/ejurnal/index.php/jsakti

Implementasi dan Analisis Attack Tree pada Aplikasi DVWA Berdasar Metrik Time dan Probability

Alfian Rifki Irawan[1], Adityas Widjajarto[2], Muhammad Fathinuddin[3]
[1,2,3]Universitas Telkom, Indonesia
e-mail: alfianrifkiirawan@student.telkomuniversity.ac.id, adtwjrt@telkomuniversity.ac.id, muhammadfathinuddin@telkomunivertity.ac.id

Abstract

The formulation of attack trees can be based on the exploitation stages in web-based applications. According to this formulation, this research aims to understand the relationship between attack trees and exploitation characteristics using time and probability metrics. The construction of attack trees is based on experimental platforms using the DVWA web-based application, both in protected and unprotected conditions by a Web Application Firewall (WAF). Exploitation is carried out on five vulnerabilities, namely SQL Injection, XSS (Reflected), Command Injection, CSRF, and Brute Force. The analysis results without a WAF show that the Cross-Site Request Forgery attack tree occupies the top position with a score of 18.19. On the other hand, the Brute Force attack tree ranks last with a score of 230.09. With the presence of a WAF, the Command Injection attack tree takes the first position with a score of 4.80, while the Brute Force attack tree remains in the last position with a score of 43.08. Further research in this study may involve a detailed examination of probability metrics and the calculation of vulnerability factors.

Keywords*: attack tree, exploitation, metrics, time, probability*

Abstrak

Perumusan attack tree dapat dilakukan berdasarkan tahapan eksploitasi pada aplikasi berbasis web. Berdasarkan rumusan tersebut, penelitian ini bertujuan untuk memahami relasi antara attack tree dan karakter eksploitasi menggunakan metrik time dan probability. Penyusunan attack tree berdasarkan platform percobaan terhadap aplikasi berbasis web DVWA dengan kondisi terlindungi dan tidak terlindungi oleh Web Application Firewall (WAF). Eksploitasi dilakukan berdasarkan lima kerentanan yaitu SQL Injection, XSS (Reflected), Command Injection, CSRF, dan Brute Force. Hasil analisis tanpa WAF menghasilkan Cross-Site Request Forgery attack tree menempati posisi pertama dengan skor 18,19. Brute Force attack tree menempati urutan terakhir dengan skor 230,09. Sedangkan dengan WAF menghasilkan Command Injection attack tree menempati posisi pertama dengan skor 4,80. Brute Force attack tree menempati urutan terakhir dengan skor 43,08. Kelanjutan penelitian ini dapat berupa rincian metrik probability dan memperhitungan faktor vulnerability.

Kata kunci*: attack tree, eksploitasi, metrik, time, probability*

1. PENDAHULUAN

Kemajuan teknologi saat ini berkembang dengan pesat yang mengakibatkan informasi dapat mudah diakses dimana saja, kapan saja, dan oleh siapa saja tanpa ada batas ruang dan waktu. Salah satu media sebagai penyebar informasi yaitu aplikasi berbasis *website*. Dengan begitu banyaknya jumlah media serta pengguna aplikasi berbasis *website*, menyebabkan muncul berbagai celah keamanan. Dari celah keamanan tersebut, dimanfaatkan oleh orang-orang yang tidak bertanggung jawab untuk mengambil keuntungan pribadi dan merugikan suatu pihak. Untuk meminimalisir serta mencegah hal-hal yang akan berdampak merugikan terhadap pengguna ataupun *developer* aplikasi berbasis web, diperlukan solusi untuk

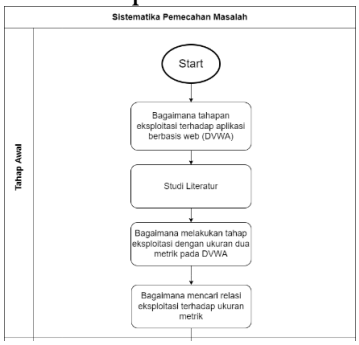
Jurnal Sains Komputer & Informatika (J-SAKTI)
Volume 7 Nomor 2, September 2023, pp. 815-827
ISSN: 2548-9771/EISSN: 2549-7200
https://tunasbangsa.ac.id/ejurnal/index.php/jsakti

mengatasinya. Salah satu solusinya yaitu dengan menggunakan *Web Application Firewall* (WAF).

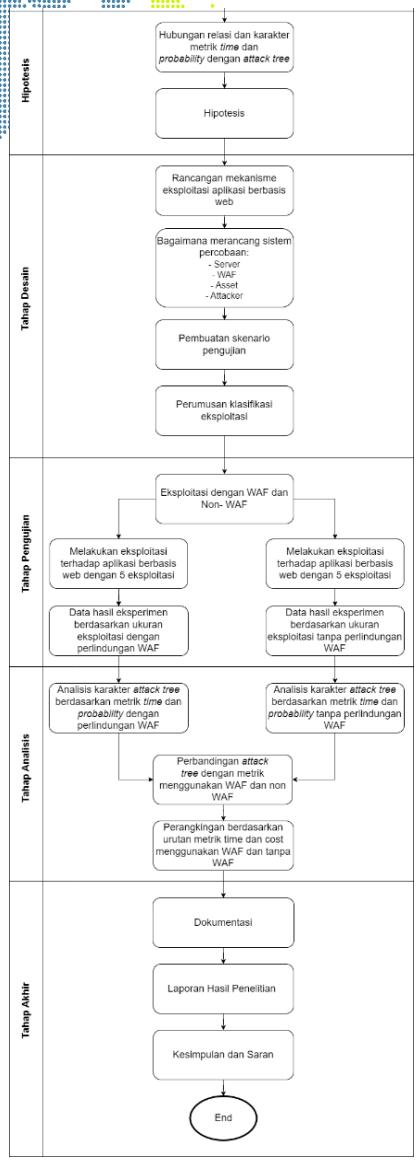
WAF merupakan *firewall* yang menjadi solusi untuk mengatasi permasalahan kerentanan pada aplikasi berbasis web. WAF memiliki kemampuan untuk melakukan *filtering* paket, memblokir lalu lintas HTTP dan juga *logging*. Terdapat berbagai macam WAF salah satunya yaitu ModSecurity[1]. Pada penelitian ini, WAF digunakan sebagai *service* untuk melindung objek yang digunakan dalam studi kasus penelitian. WAF berperan untuk melindungi objek dari berbagai serangan seperti *SQL Injection*, *Brute Force*, dan lainnya. Untuk menguji kinerja WAF, maka perlu dilakukan pengujian yaitu eksploitasi. Pada proses pengujian eksploitasi, dibutuhkan standar untuk menjadi acuan kerangka kerja penyerangan pada objek. Penelitian ini akan menggunakan daftar OWASP Top Ten sebagai acuan kerentanan kemanan yang akan dilakukan eksploitasi. Eksploitasi didasarkan pada hasil proses *vulnerability scanning* dan pengujian dilakukan dalam dua kondisi yaitu dengan kondisi perlindungan dan tidak dalam perlindungan WAF. Hasil dari pengujian eksploitasi diolah menjadi sebuah relasi tahapan eksploitasi yang digambarkan menjadi *activity diagram* dan *data flow diagram*. Kemudian, kedua data tersebut diolah menjadi sebuah kerangka penyerangan yang disebut dengan *attack tree*, yang menjelaskan tahapan eksploitasi hingga mendapatkan *gain access root*. Selain itu, data hasil dari implementasi pengujian eksploitasi dilakukan analisis yaitu berupa relasi tahapan eksploitasi berupa diagram-diagram dan menghasilkan dua pengukuran berdasarkan dua metrik yaitu metrik *time* dan metrik *probability*. Selanjutnya pada tahap analisis dilakukan perbandingan data hasil pengujian eksploitasi berdasarkan kedua metrik yang telah dibuat dari hasil proses pengujian WAF untuk mengetahui karakter dari setiap *attack tree*. Dengan begitu, *attack tree* dapat digunakan oleh penyerang untuk menentukan eksploitasi sistem dengan waktu tercepat serta kemungkinan penyerangan yang dapat terjadi.

2. METODOLOGI PENELITIAN

Sebelum dilakukannya eksperimen terhadap implementasi *Web Application Firewall* dan kemudian menghasilkan suatu data dan selanjutnya dilakukan analisis data tersebut, maka diperlukan suatu mekanisme atau sistematika penyelesaian masalah. Penyelesaian masalah dalam penelitian ini mencakup 6 tahapan diantaranya yaitu: Tahap Awal, Hipotesis, Tahap Perancangan, Tahap Eksperimen, Tahap Analisis, dan Tahap Akhir.



Jurnal Sains Komputer & Informatika (J-SAKTI)
Volume 7 Nomor 2, September 2023, pp. 815-827
ISSN: 2548-9771/EISSN: 2549-7200
https://tunasbangsa.ac.id/ejurnal/index.php/jsakti



Gambar 1. Sistematika Penyelesaian

1. Tahap Awal (Perumusan Masalah)

Tahap Awal dalam penelitian ini yaitu dengan mempelajari tahapan eksploitasi terhadap aplikasi berbasis web yaitu DVWA sebagai target eksploitasi dengan mengacu pada studi literatur. Studi literatur berguna untuk memperdalam teori mengenai tahapan eksploitasi terhadap aplikasi berbasis web DVWA melalui jurnal dan buku yang berkaitan dengan eksploitasi. Selanjutnya, dilakukan pengujian eksploitasi tanpa melakukan tahapan *post exploitation* terhadap aplikasi berbasis web DVWA untuk mendapatkan hasil eksperimen berupa metrik *time* dan *probability* dan kemudian pengujian ini menjadikan sebagai batasan masalah dalam penelitian pada tugas akhir ini. Kemudian, mendapatkan relasi eksploitasi terhadap metrik *time* dan *probability.*

2. Tahap Hipotesis

Pada tahap kedua yaitu tahap hipotesis. Pada tahapan ini melakukan hipotesis berupa praduga sementara terhadap hipotesis mengenai relasi dan karakter metrik *time* dan *probability* dengan *attack tree*.

3. Tahap Desain

Pada tahap ketiga yaitu tahap perancangan pengujian dengan melakukan perancangan mekanisme terkait eksploitasi yang dilakukan mulai dari tahap perancanaan dan persiapan. Selanjutnya melakukan installasi software pada virtual machine dan server yang terdiri dari:

a) VM Ubuntu Server sebagai server dan
b) VM Kali Linux sebagai penyerang

Setelah itu, dilakukan pembuatan skenario pengujian dimulai dari *vulnerability scanning* hingga eksploitasi. Hasil dari *vulnerability scanning* dilakukan klasifikasi jenis penyerangan sesuai dengan standar yang digunakan. Langkah selanjutnya yaitu melakukan percobaan eksploitasi yang akan dilakukan pada tahap pengujian selanjutnya.

4. Tahap Pengujian

Pada tahap ke empat yaitu tahap pengujian yang terdiri dari tahapan proses eksploitasi dengan kondisi:

a) Eksploitasi dilakukan dengan mematikan WAF
b) Eksploitasi dilakukan dengan menyalakan WAF

Eksploitasi yang dilakukan terhadap aplikasi berbasis web berdasarkan ketetapan lima eksploitasi yang telah ditentukan pada tahap sebelumnya. Bersamaan dengan pengujian tersebut, dilakukan pencatatan data hasil eksperimen pengujian eksploitasi terhadap aplikasi berbasis web DVWA, yaitu:

a) Data hasil eksperimen eksploitasi dengan mematikan WAF
b) Data hasil eksperimen eksploitasi dengan menyalakan WAF

Sehingga didapatkan data hasil eksploitasi berupa *time* dan *probability*. Setelah mendapatkan hasil dari melakukan pengujian eksploitasi, dilakukan analisis terhadap metrik dan *attack tree* pada tahap selanjutnya.

5. Tahap Analisis

Pada tahap ke lima yaitu melakukan analisis. Dilakukan proses analisis terhadap data hasil eksperimen pengujian yang telah dilakukan pada tahap pengujian. Analisis dilakukan untuk mengetahui karakteristik *attack tree* dari metrik *time* dan *probability* berdasarkan eksploitasi pada skenario pengujian dengan kondisi:

a) Pengujian eksploitasi dengan perlindungan WAF.
b) Pengujian eksploitasi tanpa perlidungan WAF.

Selain itu, analisis dilakukan untuk mengukur eksploitasi yang menghasilkan data metrik time dan probability. Hasil analisis ini menunjukkan karakteristik *attack tree* dari metrik *time* dan *probability* yang memiliki relasi dengan eksploitasi. Selanjutnya hasil analisis yang sudah didapatkan akan dijadikan sebagai perbandingan dan digunakan untuk penyusunan pola *attack tree* berdasarkan kondisi aplikasi berbasis web dengan perlindungan dan tanpa perlindungan WAF. Kemudian, dilakukan perangkingan berdasarkan urutan metrik time dan

Jurnal Sains Komputer & Informatika (J-SAKTI)
Volume 7 Nomor 2, September 2023, pp. 815-827
ISSN: 2548-9771/EISSN: 2549-7200
https://tunasbangsa.ac.id/ejurnal/index.php/jsakti

probability terhadap aplikasi berbasis web dalam kondisi dengan menggunakan WAF atau tidak menggunakannya.

6. Tahap Akhir

Pada tahap akhir ini, berupa penyusunan kesimpulan berdasarkan nilai tertinggi metrik *time* dan *probability* untuk *attack tree*, saran yang diperoleh berdasarkan pengujian eksploitasi terhadap aplikasi berbasis web DVWA

3. HASIL DAN PEMBAHASAN

3.1. *Reconnaissance*

Reconnaissance merupakan sebuah fase persiapan sebelum (*attacker*) melakukan pencurian informasi pada web *server*, di mana kegiatan ini adalah untuk mengumpulkan informasi sebanyak mungkin mengenai sasaran web *server*.[10] Kumpulan informasi tersebut menjadi sebuah panduan seorang penguji untuk melakukan serangan kepada web *server* yang dituju

3.2. Spesifikasi Hardware dan Software

Dalam proses upaya pengujian eksploitasi terhadap suatu sistem, diperlukan *hardware* dan *software* untuk menunjang dalam proses pengambilan data ketika dilakukan eksploitasi terhadap aplikasi berbasis web. Berikut merupakan *hardware* dan *software* yang digunakan dalam proses pengujian eksploitasi yang tertera pada Tabel 1 dan Tabel 2:

Tabel 1. Spesifikasi *Hardware*

Komponen	Informasi	
Spesifikasi *Server*	*Processor*	Intel® Pentium® Gold G5400 CPU @4.00GHz (2CPUs) TDP 56W
	Memory	20393 MB DDR4 LONGDIMM 2666 MHz
	Hard Disk	120 GB SSD
	System Type	64-Bit
	Operating System	Linux Ubuntu 22.04 LTS
Spesifikasi *Main OS*	*Processor*	Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz (12 CPUs), ~2.6GHz
	Memory	16384MB RAM
	Hard Disk	2 TB
	System Type	64-bit
	Operating System	Windows 11 Home Single Language (10.0, Build 22000)
Spesifikasi *Virtual Machine*	*Processor*	3 *Processor*
	Memory	5084MB RAM
	Hard Disk	60 GB
	System Type	64-bit
	Operating System	Kali Linux 2023.1 Kali-rolling

Tabel 2. Spesifikasi *Software*

Tipe	Software	Versi
Operating System	Kali Linux	2023.1 Kali-rolling

Jurnal Sains Komputer & Informatika (J-SAKTI)
Volume 7 Nomor 2, September 2023, pp. 815-827
ISSN: 2548-9771/EISSN: 2549-7200
https://tunasbangsa.ac.id/ejurnal/index.php/jsakti

Tipe	Software	Versi
Web Application	DVWA	2023
Web Application Firewall	ModSecurity	3.3.2
Attack Tools	Sqlmap	1.7.2
	Wfuzz	3.1.0
	Burp Suite	2023.1.2
	Firefox	102.8.0esr (64-bit)
Vulnerability Scanning	OWASP - ZAP	2.12.0

Berdasarkan Tabel 2, disebutkan spesifikasi yang digunakan selama penelitian dan pengujian, selanjutnya pada bagian ini akan dijelaskan mengenai fungsi-fungsi dari setiap perangkat lunak yang telah disebutkan pada Tabel IV.2, yaitu sebagai berikut:

1. *Operating System*

Kali Linux adalah distro Linux yang berisi ratusan koleksi perangkat lunak yang secara khusus dirancang untuk pengguna melakukan penetration testing dan kemanan professional lainnya[2]. Kali Linux dikembangkan oleh O*ffensive Security* yang merupakan perusahaan internasional Amerika yang bergerak dibidang forensika digital, kemananan informasi, serta pengujian penetrasi.

2. *Web Application*

DVWA adalah sebuah *tools* yang dirancang untuk membantu *security professional* dan pengembang web untuk melakukan pengujian dan menguji keterampilan *skill* dan *tools* yang digunakan dalam lingkup praktik hukum[3]. DVWA dirancang khusus dengan memiliki berbagai macam kerentanan pada sistemnya yang bertujuan untuk dilakukan pengujian serta untuk mengasah kemampuan *penetration testing*. Kerentanan sistem keamanan DVWA dapat diatur sesuai *level* yang kita pilih diantaranya *low, medium, high,* dan *impossible.*

3. *Web Application Firewall*

ModSecurity adalah *software* yang umum digunakan untuk pelacakan, logging dan manajemen aplikasi web secara *realtime*[4]. ModSecurity bekerja sebagai *firewall* yang tugasnya untuk melindungi aplikasi web dari berbagai macam serangan dan ancaman keamanan dengan mendeteksi *request* yang bersifat anomali, kemudian dapat membuat pencatatan ke dalam bentuk log, serta dapat melakukan penyaringan terhadap *request* HTTP berdasarkan aturan atau rule yang telah dikonfigurasi dinamakan dengan SecRule.

4. *Attack Tools*

SQLMap adalah aplikasi open source atau tool yang terdapat dalam Kali Linux[5]. Sqlmap digunakan untuk *penetration testing* dan secara otomatis mendeteksi serta melakukan eksploitasi dengan *SQL Injection*. Pada proses implementasinya, Sqlmap membutuhkan URL HTTP *request* target yang bersifat rentan, sehingga dapat dilakukan eksploitasi terhadap *database website* yang akhirnya bisa mendapatkan informasi tentang struktur *database* diantaranya nama database, isi dari tabel database, isi dari kolom database, hingga data yang tersedia dalam kolom database.

Wfuzz merupakan salah satu fuzzer yang dibangun dengan bahasa python dan terdiri dari banyak komponen yang dapat dikembangkan[6]. Tujuan utama

Jurnal Sains Komputer & Informatika (J-SAKTI)
Volume 7 Nomor 2, September 2023, pp. 815-827
ISSN: 2548-9771/EISSN: 2549-7200
https://tunasbangsa.ac.id/ejurnal/index.php/jsakti

dari aplikasi wfuzz ini adalah melakukan serangan *fuzzing* kepada aplikasi web. Pada prosesnya, fuzzing melakukan serangkaian HTTP *request* dan secara otomatis nilai parameter akan berubah. Wfuzz berusaha mengirimkan *request* dengan variasi nilai parameter yang berbeda-beda untuk menemukan celah keamanan dari aplikasi web. Selain itu, Wfuzz mendukung penggunaan payload yang dapat disesuaikan dengan kebutuhannya, kemudian menggunakan ekresi regular (regex), serta menghasilkan suatu laporan hasil pengujian yang berisikan detail tentang respon *server*, kode status HTTP, dan lainnya.

Burpsuite adalah sebuah *software* lengkap untuk melakukan pengujian kemanan aplikasi web yang dikembangkan oleh *Port*swigger. Cara kerja burpsuite yaitu dengan fitur proxy. Dengan fitur ini, pengguna dapat mencatat, mencegat, menampilkan serta dapat memodifikasi lalu lintas HTTP antara browser dengan *server* [7].

Firefox adalah *open source web browser* yang dikembangkan oleh Mozilla Corporation[8]. Firefox merupakan *web browser* populer yang sudah beralih dari model pengembangan tradisional ke model *rapid release*. Firefox berfungsi sebagai alat untuk memuat dan menampilkan halaman web secara visual yang menarik sehingga pengguna mudah memahami isi dari halaman web tersebut. Selain itu juga, firefox dapat melakukan perubahan *script* pada *website* yang akan dilakukan pengujian keamanan dan firefox juga dapat mengindentifikasi lalu lintas *request* dan *post* HTTP pada *website* sehingga menghasilkan informasi yang dapat digunakan untuk melakukan *penetration testing*.

5. *Vulnerability Scanning*

OWASP ZAP (Zed Attack Proxy) merupakan sebuah aplikasi untuk melakukan *penetration testing* dalam menemukan *vulnerabilities*/celah keamanan pada suatu aplikasi web . ZAP menyediakan *scanner* secara automatis[9].

3.3. *Scanning*

Pada tahap penelitian ini, akan dilakukan scanning dengan menggunakan OWASP-ZAP terhadap aplikasi berbasis web DVWA yang kemudian menghasilkan *output report*. Dari hasil *scanning* tersebut dilakukan batasan *vulnerability* yang akan dilakukan eksploitasi pada tahap selanjutnya. Berikut merupakan hasil *scanning*

Tabel 3. Hasil Pengujian *Vulnerability Scanning* Menggunakan OWASP-ZAP

Site	Risk			
	High (= High)	**Medium (>= Medium)**	**Low (>= Low)**	**Informational (>= Informational)**
http://172.28.232.111	8	11	8	7
	8	19	27	34

Berdasarkan Tabel 3 diketahui pada aspek *risk* memiliki *level high* sebesar 8 kerentanan, pada *level medium* sebesar 11 kerentanan, *level low* sebesar 8 kerentanan, dan pada *level informational* ada sebesar 7 kerentanan. Sehingga total dari keseluruhan memiliki jumlah sebesar 34 kerentanan. Dari sekian banyak

Jurnal Sains Komputer & Informatika (J-SAKTI)
Volume 7 Nomor 2, September 2023, pp. 815-827
ISSN: 2548-9771/EISSN: 2549-7200
https://tunasbangsa.ac.id/ejurnal/index.php/jsakti

kerentanan yang berhasil didapat, pada penelitian ini dibatasi jumlah serangan yang akan eksploitasi yaitu pada tabel berikut :

Tabel 4. Hasil Pengujian *Vulnerability Scanning* Menggunakan OWASP-ZAP yang Akan Dieksploitasi

Deskripsi	***Risk Level***	**CWE ID**	**WASC ID**	**Alert ID**
Cross Site *Script*ing (Reflected)	*High*	79	8	Active (40012 - Cross Site *Script*ing (Reflected))
SQL Injection	*High*	89	19	Active (40018 - SQL Injection)
Remote OS Command Injection	*High*	78	31	Active (90020 - Remote OS Command Injection)
Absence of Anti-CSRF Tokens	*Medium*	352	9	Passive (10202 - Absence of Anti-CSRF Tokens)
User Agent Fuzzer // Brute Force	*Informational*	0	0	Active (10104 - User Agent Fuzzer)

3.4. Analisis Perbandingan Metrik *Time*

Pengukuran *time* yang dilakukan berdasarkan berapa lama proses eksploitasi terhadap aplikasi berbasis web dilakukan sesuai dengan langkah-langkah yang telah di tentukan. Pengukuran *time* dilakukan dalam nilai detik (*s*). Waktu ini kemudian dibagi menjadi tiga aspek yang berbeda, yaitu *real time*, *user time*, dan *system time.* Berikut merupakan tabel yang menunjukkan metrik *time* dari eksploitasi yang dilakukan dengan kondisi aplikasi berbasis web tidak dilindungi oleh WAF dan dilindungi oleh WAF

Tabel 5. Analisis Perbandingan Metrik *Time* tanpa WAF

Number	**Serangan**	**Time (s)**				
		Real	**User**	**Sys**	**Total**	**CPU**
1	*Attack Tree SQL Injection*	1.084,70	17,32	4,12	1.106,14	94%
2	*Attack Tree Cross-Site Scripting* (*Reflected*)	39,92	16,04	4,31	60,27	70%
3	*Attack Tree Command Injection*	91,12	12,84	12,45	116,41	26%
4	*Attack Tree Cross Site Request Foreign*	72,73	20,66	5,91	99,3	79%
5	*Attack Tree Brute Force*	920,37	613,91	187,55	1.721,83	55%

Tabel 6. Analisis Perbandingan Metrik *Time* dengan WAF

Number	**Serangan**	**Time (s)**				
		Real	**User**	**Sys**	**Total**	**CPU**
1	*Attack Tree SQL Injection*	41,76	15,37	13,29	70,42	64%
2	*Attack Tree Cross-Site Scripting* (*Reflected*)	50,25	11,51	12,11	73,87	62%
3	*Attack Tree Command Injection*	63,99	11,86	14,32	90,17	35%
4	*Attack Tree Cross Site Request Foreign*	78,25	15,45	17,24	110,94	74%
5	*Attack Tree Brute Force*	750,47	304,16	25,47	1.080,10	63%

Jurnal Sains Komputer & Informatika (J-SAKTI)
Volume 7 Nomor 2, September 2023, pp. 815-827
ISSN: 2548-9771/EISSN: 2549-7200
https://tunasbangsa.ac.id/ejurnal/index.php/jsakti

Pengukuran metrik *time* dicacat menjadi tiga aspek bagian yaitu: *Real*, *User*, dan *System*. Pada penelitian ini akan terbatas *time* pada aspek *Real*. Dikarenakan aspek *Real* sudah mewakili dari kedua aspek lainnya. Metrik *time* perlu dilakukan perhitungan. Berikut merupakan rumus dalam menghitung metrik *time* pada setiap eksploitasi:

$$\sum_{i=1}^{n} t(A) = r_1 + r_2 + \cdots + r_n \ \ldots . (i) \quad (1)$$

Dengan:
$t(A) = Attack\ time\ (detik)$
$n =$ batas atas
$i =$ indeks penjumlahan
$r = real\ time$

Setelah dilakukannya perhitungan terhadap metrik *time*, selanjutnya yaitu melakukan identifikasi terhadap metrik time dari setiap eksploitasi yang dilakukan dengan kondisi perlindungan atau tanpa perlindungan WAF:

Tabel 7. Perbandingan Metrik Time antar Eksploitasi tanpa WAF

Attack Tree	***Time* Metrik (*s*)**
Attack Tree SQL Injection	1.084,70
Attack Tree Cross-Site Scripting (*Reflected*)	39,92
Attack Tree Command Injection	91,12
Attack Tree Cross Site Request Foreign	72,73
Attack Tree Brute Force	920,37

Tabel 8. Perbandingan Metrik *Time* Antar Eksploitasi dengan WAF

Attack Tree	**Metrik *time*(*s*)**
Attack Tree SQL Injection	41,76
Attack Tree Cross-Site Scripting (*Reflected*)	50,25
Attack Tree Command Injection	63,99
Attack Tree Cross Site Request Foreign	78,25
Attack Tree Brute Force	750,47

3.5. Analisis Perbandingan Metrik *Probability*

Pengukuran *probability* dilakukan berdasarkan penyerangan yang mengacu kepada tahapan eksploitasi yang memiliki nilai keberhasilan suatu *step* eksploitasi dilakukan. Pengukuran dilakukan perhitungan dengan menggunakan rumus berikut:

$$\sum_{i=1}^{n} p(A) = x_1 + x_2 + \cdots + x_n \quad \ldots . (i) \quad (2)$$

$p(A) = Probability\ Attack$
$x =\ Probability$ Step Eksploitasi

Berikut merupakan tabel yang menunjukkan persentase *probability* keberhasilan suatu tahapan eksploitasi terhadap aplikasi berbasis web dengan kondisi tidak terlindungi dan dengan terlindungi oleh WAF:

Tabel 9. Perbandingan Metrik *Probability* Antar Eksploitasi Tanpa WAF

Eksploitasi	***Probability* Metrik (%)**
Attack Tree SQL Injection	100%

Jurnal Sains Komputer & Informatika (J-SAKTI)
Volume 7 Nomor 2, September 2023, pp. 815-827
ISSN: 2548-9771/EISSN: 2549-7200
https://tunasbangsa.ac.id/ejurnal/index.php/jsakti

Eksploitasi	*Probability* Metrik (%)
Attack Tree Cross-Site Scripting (Reflected)	100%
Attack Tree Command Injection	100%
Attack Tree Cross Site Request Foreign	100%
Attack Tree Brute Force	100%

Tabel 10. Perbandingan Metrik *Probability* antar Eksploitasi dengan WAF

Eksploitasi	*Probability* Metrik (%)
Attack Tree SQL Injection	28,56%
Attack Tree Cross-Site Scripting (Reflected)	50%
Attack Tree Command Injection	50%
Attack Tree Cross Site Request Foreign	75%
Attack Tree Brute Force	75%

3.6. Analisis Perbandingan *Attack Tree* Berdasarkan Metrik *Time* dan *Probability*

Dalam menentukan perangkingan terhadap kedua metrik yaitu metrik *time* dan *probability*, diperlukan suatu rumus untuk melakukan perhitungan yang kemudian menghasilkan sebuah nilai atau angka yang menjadi panduan dalam perangkingan suatu pengujian eksploitasi. Berikut merupakan rumus perhitungan untuk perbandingan antar metrik:

$$\sum_{i=1}^{n} \mathrm{Q}(A) = (r_1.x_1) + (r_2.x_2), + \cdots (r_n.x_n) \quad (3)$$

Dengan:
$Q(A)$ = *Time Probability Score*
r = *real time* yang dibutuhkan pada penyerangan
x = *probability* yang dihasilkan pada penyerangan

Setelah dilakukan perhitungan dengan menggunakan rumus tersebut, diperoleh suatu kalkulasi data yang dinamakan sebagai *Time Probability Score*. Nilai dari *time probability score* dapat digunakan dalam pengurutan perangkingan eksploitasi berdasarkan skor paling rendah hingga skor paling tinggi. Berikut merupakan tabel yang menampilkan hasil pengurutan perangkingan dengan *time probability score* dengan kondisi tidak dalam perlindungan dan dalam perlindungan WAF:

Tabel 11. Perangkingan *Time* dan *Probability* Metrik pada Pengujian Ekploitasi tanpa Perlindungan WAF

Rank	Eksploitasi	*Time Probability Score*
1	*Cross-Site Request Forgery*	18,18
2	*Cross-Site Scripting (Reflected)*	19,96
3	*Command Injection*	22,78
3	*SQL Injection*	154,90
4	*Brute Force*	230,09

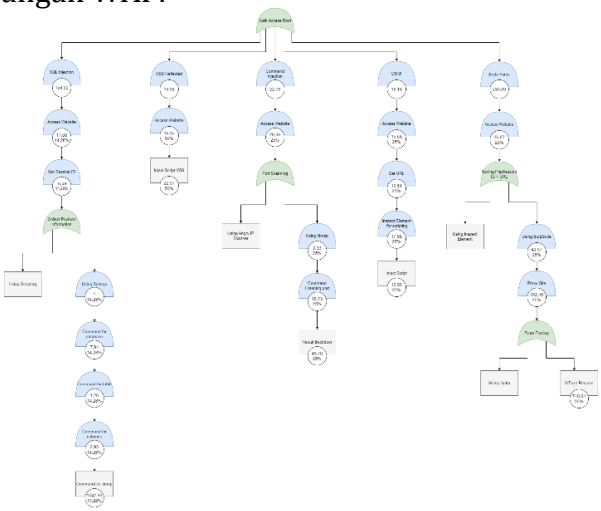
Jurnal Sains Komputer & Informatika (J-SAKTI)
Volume 7 Nomor 2, September 2023, pp. 815-827
ISSN: 2548-9771/EISSN: 2549-7200
https://tunasbangsa.ac.id/ejurnal/index.php/jsakti

Pada Tabel 11 menampilkan bahwa eksploitasi *Cross-Site Request Forgery* berada pada posisi urutan pertama dengan skor 18.18, sementara eksploitasi *Brute Force* berada pada posisi terakhir dengan skor 230;09.

Tabel 12. Perangkingan *Time* dan *Probability* Metrik pada Pengujian Ekploitasi dengan Perlindungan WAF

Rank	Eksploitasi	Time Probability Score
1	*Command Injection*	4,79
2	*SQL Injection*	5,09
3	*Cross-Site Scripting (Reflected)*	8,48
4	*Cross-Site Request Forgery*	16,45
5	*Brute Force*	43,08

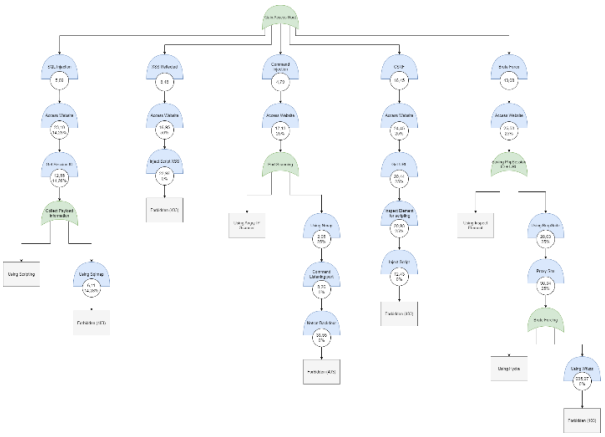
Pada Tabel 12 menampilkan bahwa eksploitasi *Command Injection* berada pada posisi urutan pertama dengan skor 4,79, sementara eksploitasi *Brute Force* berada pada posisi dengan skor 43,08. Setelah mendapatkan urutan ranking pada keseluruhan eksploitasi, kemudian hasil akhir dibuatkan visualisasi *attack tree* yang berisikan *time probability score* serta *time* dan *probability* pada setiap *node attack tree* yang merepresentasikan tahapan pada eksploitasi. Berikut merupakan attack tree dengan metrik *time* dan *probability* dengan kondisi tanpa perlindungan dan dengan perlindungan WAF:



Gambar 2. Diagram *Attack Tree* dengan Metrik *Time* dan *Probability* tanpa Perlindungan WAF

Pada Gambar 2 menjelaskan tentang diagram *attack tree development* dari semua eksploitasi. Tujuan dari semua eksploitasi tersebut adalah untuk mendapatkan *gain access root* secara ilegal. Dalam diagram attack tree tersebut memuat lima eksploitasi yaitu, *SQL Injection*, XSS (*Reflected*), *Command Injection*, CSRF, dan *Brute Force*. Setiap *node attack tree* mengandung langkah-langkah eksploitasi yang berisikan metrik *time* dan *probability* dengan kondisi tanpa perlindungan WAF. Nilai waktu pada setiap langkah dalam *attack tree* adalah

waktu yang dihabiskan ketika pengujian eksploitasi dilakukan hingga satu tahap eksploitasi tersebut selesai. Ukuran *probability* pada setiap langkah dalam *attack tree* adalah peluang keberhasilan suatu tahap eksploitasi ketika pengujian eksploitasi dilakukan. Sehingga, skor untuk perangkingan dapat dilihat pada nama eksploitasi.



Gambar 3. Diagram *Attack Tree* dengan Metrik *Time* dan *Probability* dengan Perlindungan WAF

Pada Gambar 3 menjelaskan tentang diagram *attack tree development* dari semua eksploitasi. Tujuan dari semua eksploitasi tersebut adalah untuk mendapatkan *gain access root* secara ilegal. Dalam diagram *attack tree* tersebut memuat lima eksploitasi yaitu, *SQL Injection*, XSS (*Reflected*), *Command Injection*, CSRF, dan *Brute Force*. Setiap *node attack tree* mengandung langkah-langkah eksploitasi yang berisikan metrik *time* dan *probability* dengan kondisi dengan perlindungan WAF. Nilai waktu pada setiap langkah dalam *attack tree* adalah waktu yang dihabiskan ketika pengujian eksploitasi dilakukan hingga satu tahap eksploitasi tersebut selesai. Ukuran *probability* pada setiap langkah dalam *attack tree* adalah peluang keberhasilan suatu tahap eksploitasi ketika pengujian eksploitasi dilakukan. Sehingga, skor untuk perangkingan dapat dilihat pada nama eksploitasi. Salah satu bentuk perlindungan yang dilakukan oleh WAF adalah dengan memutus koneksi dari penyerang yang kemudian memberikan pesan berupa kode atau halaman yang menampilkan “403 Forbidden”.

4. SIMPULAN

Berdasarkan analisa pada bagian sebelumnya, penelitian ini menghasilkan kesimpulan bahwa *Attack tree* dapat disusun berdasarkan hasil tahapan eksploitasi yang digambarkan dengan *activity diagram* dan *data flow diagram*. Karakter *attack tree* dapat disusun menggunakan relasi metrik *time* dan *probability*. Kedua metrik tersebut dapat digunakan untuk perangkingan berbagai *attack tree*. Perangkingan tertinggi tanpa WAF adalah CSRF dengan skor 18,18. *Brute Force* menempati urutan terakhir dengan skor 230,09. Dengan peranan WAF yang memblokir eksploitasi, berpengaruh pada perangkingan *attack tree*. Dengan skor tertinggi *Command Injection* 4,80. *Brute Force* menempati urutan terakhir dengan skor 43,08.

DAFTAR PUSTAKA

[1] Agung Muzaki, R., Ritchi, H., Candra Briliyant, O., & Andika Hasditama, M. (2020). Improving Security of Web-Based Application Using ModSecurity and Reverse Proxy in Web Application Firewall.

[2] Hertzog, R., O'Gorman, J., & Aharoni, M. (n.d.). Kali Linux revealed : mastering the penetration testing distribution.

[3] Abdoulaye Kindy, D., & Khan Pathan, A.-S. (n.d.). A Detailed Survey on Various Aspects of SQL Injection in Web Applications: Vulnerabilities, Innovative Attacks, and Remedies. In *International Journal DRAFT*.

[3] Zavarsky Sergey Butakov, P., David Sobola, T., Supervisor Edgar Schmidt, P., Schmidt, E., Dean, Ds., Zavarsky, P., & Butakov, S. (2020). *Experimental Study Of Modsecurity Web Application Firewalls Co-Authored By Timilehin David Sobola Experimental Study Of Modsecurity Web Application Firewalls Experimental Study of ModSecurity Web Application Firewalls*.

[4] Lika, S., Dwi, R., Halim, P., & Verdian, I. (2018). *Positif : Jurnal Sistem dan Teknologi Informasi Analisa Serangan Sql Injeksi Menggunakan SQLMAP Implementation Of Online Accounting Software As Supporting Of Financial Statement*. *4*(2).

[5] Yogi Kristiawan, O., & Teknologi Bandung Menyetujui Pembimbing, I. (2017). *Perancangan Dan Implementasi Rule Based Dictionary Attack Pada Fuzzer Wfuzz Untuk Menguji Kerentanan Aplikasi Web (Program Studi Magister Teknik Elektro)*.

[6] Mainka, C., Mladenov, V., Guenther, T., & Schwenk, J. (2015). *Automatic Recognition, Processing and Attacking of Single Sign-On Protocols with Burp Suite*. https://github.com/RUB-NDS/BurpSSOExtension.

[7] Khomh, F., Dhaliwal, T., Zou, Y., & Adams, B. (n.d.). *Do Faster Releases Improve Software Quality?-An Empirical Case Study of Mozilla Firefox*.

[8] Saputra, A., Armys Roma Sitorus, M., & Negeri Batam Program Studi Teknik Multimedia dan Jaringan Jalan Ahmad Yani, P. (2017). Penilaian Ancaman pada Website Transkrip Aktivitas Kemahasiswaan Politeknik Negeri Batam Menggunakan Metode DREAD. In *Jurnal Integrasi* (Vol. 9, Issue 1). http://www.tak.polibatam.ac.id.