

Implementasi dan Analisis Attack Tree pada Aplikasi DVWA Berdasar Metrik Time dan Skill Level

1st Yadi Nugraha
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia
yadinugraha@student.telkomuni-
versity.ac.id

2nd Adityas Widjajarto
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia
adtwjrt@telkomuniversity.ac.id

3rd Muhammad Fathinuddin
Fakultas Rekayasa Industri
Universitas Telkom
Bandung, Indonesia
muhammadfathinuddin@telkom-
university.ac.id

Abstrak - Attack tree dapat dirumuskan berdasarkan langkah-langkah eksploitasi yang terjadi pada aplikasi web. Tujuan dari penelitian ini adalah untuk memahami relasi attack tree dan karakter eksploitasi berdasarkan metrik time dan skill level. Platform untuk pengujian eksploitasi menggunakan DVWA dan disusun menjadi attack tree. Penyusunan attack tree dengan kondisi terlindungi dan tidak terlindungi WAF. Attack tree disusun berdasarkan lima kerentanan yaitu SQL Injection, XSS (Reflected), Command injection, CSRF, dan Brute force. Hasil analisis dengan kondisi tidak dilindungi WAF menyimpulkan XSS (Reflected) attack tree menempati urutan pertama dengan skor 131,92. SQL Injection attack tree menempati urutan terakhir dengan skor 1727,56. Sedangkan dengan WAF SQL Injection attack tree menempati urutan pertama dengan skor 54. Brute force attack tree menempati urutan terakhir dengan skor 319,51. Kelanjutan penelitian dapat berupa merinci langkah eksploitasi menggunakan CVSS score sebagai perhitungan skill level dan pengukuran parameter menggunakan IDS sebagai salah satu fitur firewall.

Kata kunci: attack tree, eksploitasi, metrik, time, skill level

I. PENDAHULUAN

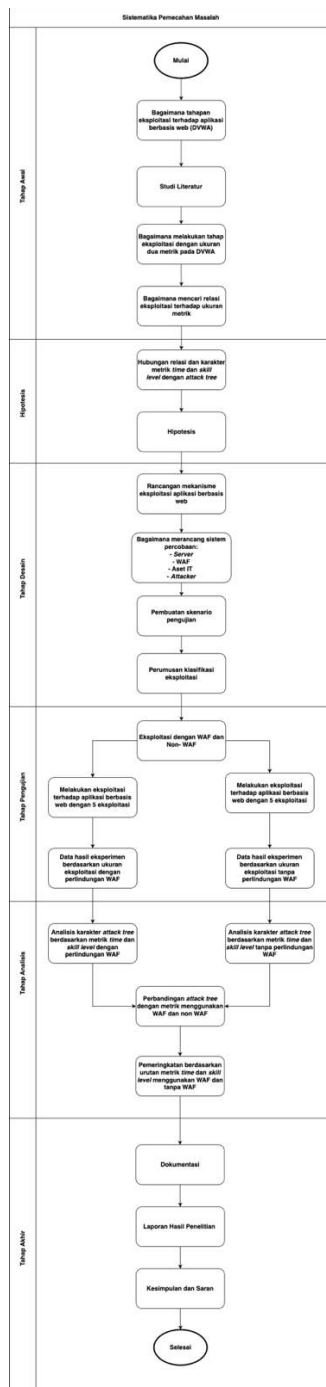
Dengan meningkatnya jumlah pengguna aplikasi berbasis *web* ini, banyak pengembang yang tidak memperhatikan keamanan aplikasi *web* dengan sebaik-baiknya. Selain itu, dengan perkembangan aplikasi berbasis *web*, muncul pula oknum-oknum yang ingin mengambil keuntungan pribadi yang pada umumnya disebut *cybercrime*. Kebutuhan akan keamanan siber menjadi prioritas saat ini, sehingga ada kebutuhan untuk menganalisis dan mengevaluasi beberapa dari ribuan kemungkinan serangan siber untuk menciptakan pertahanan di dunia siber. Salah satu solusinya adalah menggunakan *Web Application Firewall* (WAF) berfungsi untuk menolak paket data yang dianggap mencurigakan dan melakukan pencatatan aktifitas [1].

Penyerang memiliki berbagai cara untuk memanfaatkan kerentanan keamanan dalam sistem komputer atau jaringan, memungkinkan mereka menemukan jalur tercepat untuk melakukan eksploitasi. Salah satu metode untuk mengidentifikasi celah keamanan pada aplikasi *website* adalah dengan menggunakan *attack tree*. *Attack tree* digunakan untuk membuat perhitungan *metrics* diantaranya yaitu waktu, biaya, frekuensi, kemungkinan untuk berhasil menyerang, tingkat kemampuan yang dibutuhkan untuk melakukan penyerangan, peralatan khusus untuk melakukan penyerangan, dan kombinasi dari semua [2].

Penelitian ini akan melakukan eksploitasi terhadap aplikasi *web* DVWA dengan menggunakan standar dari OWASP TOP TEN yang menjadi acuan kerangka penyerangan terhadap aplikasi *web* DVWA sebagai target eksploitasi serta implementasi peran kinerja WAF terhadap pengujian eksploitasi dengan tujuan untuk menganalisa dan menyusun relasi eksploitasi berdasarkan data eksperimen yang akan dilakukan. Metode serangan eksploitasi yang dipakai dalam pengujian berdasarkan hasil *vulnerability scanning* dan pengujian eksploitasi dilakukan dengan dua kondisi yaitu pada saat aplikasi *web* dalam perlindungan WAF dan tanpa perlindungan WAF. Selanjutnya akan dilakukan penyusunan *attack tree* berdasarkan data eksperimen. Pada analisis akan dilakukan perbandingan hasil data pengujian eksploitasi berdasarkan metrik. Hasil dari analisis pengujian bertujuan untuk mengetahui karakter eksploitasi dan pengurutan eksploitasi berdasarkan metrik metrik yang diukur pada pengujian eksploitasi menggunakan *attack tree*.

II. METODOLOGI PENELITIAN

Diperlukan sistematika penyelesaian masalah yaitu tahapan yang akan dilakukan dari pengerjaan penelitian di tahap awal sampai tahap terakhir. Penyelesaian masalah penelitian ini mencakup 6 tahapan, yaitu: Tahap Awal, Hipotesis, Tahap Perancangan, Tahap Eksperimen, Tahap Analisis, dan Tahap Akhir.



Gambar 1. Sistematika Penyelesaian

1. Tahap Awal

Pada penelitian ini dimulai dengan mempelajari tahapan eksploitasi terhadap aplikasi berbasis *web* DVWA sebagai target untuk eksploitasi mengacu pada studi literatur. Studi literatur berguna untuk memperdalam teori mengenai eksploitasi terhadap aplikasi berbasis *web* (DVWA) melalui jurnal dan buku yang berkaitan dengan eksploitasi. Selanjutnya, dilakukan pengujian eksploitasi tanpa melakukan *post exploitation* terhadap aplikasi *web* DVWA untuk mendapatkan hasil eksperimen metrik *time* dan *skill level*, pengujian ini dijadikan sebagai Batasan masalah dalam penelitian pada tugas akhir ini. Kemudian,

mendapatkan relasi eksploitasi terhadap metrik *time* dan *skill level*.

2. Tahap Hipotesis

Pada tahap kedua yaitu tahap hipotesis. Pada tahap ini melakukan hipotesis terkait praduga sementara terhadap hipotesis mengenai relasi dan karakteristik metrik *time* dan *skill level* dengan *attack tree*.

3. Tahap Desain

Pada tahap desain, yaitu tahap perancangan pengujian dengan dilakukannya perancangan mekanisme pengujian eksploitasi yang dimulai dari perencanaan dan persiapan. Selanjutnya melakukan instalasi *software* pada *virtual machine* dan *server* yang terdiri dari:

1. VM Ubuntu sebagai server
2. VM Kali Linux sebagai penyerang

Selanjutnya dilakukan pembuatan skenario pengujian dimulai dari *vulnerability scanning* hingga eksploitasi. Kemudian melakukan klasifikasi jenis penyerangan sesuai dengan hasil *vulnerability scanning* dan melakukan langkah percobaan eksploitasi yang akan dilakukan pada tahap pengujian selanjutnya.

4. Tahap Pengujian

Pada tahap ini akan dilakukan tahapan eksploitasi terhadap lima eksploitasi yang terpilih dengan dua kondisi, yaitu sebagai berikut

1. Melakukan eksploitasi tanpa perlindungan WAF
2. Melakukan eksploitasi dengan perlindungan WAF

Bersamaan dengan pengujian tersebut dilakukan pencatatan data hasil eksperimen pengujian eksploitasi terhadap aplikasi *web* DVWA, yaitu

1. Data hasil eksperimen eksploitasi tanpa perlindungan WAF
2. Data hasil eksperimen eksploitasi dengan perlindungan WAF

Sehingga akan menghasilkan data hasil eksploitasi berupa *time* dan *skill level*. Selanjutnya setelah mendapatkan hasil dari melakukan pengujian eksploitasi, dilakukan analisis dan pembuatan *attack tree* pada tahap selanjutnya.

5. Tahap Analisis

Pada tahap 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 *skill level* yang digambarkan secara visual dalam bentuk *attack tree* dengan dua kondisi yaitu:

1. Pengujian eksploitasi dengan perlindungan WAF.
2. Pengujian eksploitasi tanpa perlindungan WAF.

Selain itu, analisis dilakukan untuk mengukur eksploitasi yang menghasilkan data metrik *time* dan *skill level*. Hasil dari analisis ini menunjukkan karakter dari *attack tree* pada metrik yang memiliki relasi dengan eksploitasi. Selanjutnya hasil analisis yang sudah didapatkan akan dijadikan sebagai perbandingan dan digunakan untuk penyusunan *attack tree* berdasarkan kondisi aplikasi *web* DVWA dengan perlindungan WAF dan tanpa perlindungan WAF. Kemudian, dilakukan pemeringkatan berdasarkan urutan metrik *time* dan *skill level* terhadap aplikasi *web* DVWA.

6. Tahap Akhir

Pada tahap akhir ini, berupa penyusunan kesimpulan terkait dengan hasil pengujian eksploitasi yang menghasilkan data akhir untuk analisis karakter *attack tree* berdasarkan metrik *time* dan *skill level*, saran yang diperoleh berdasarkan pengujian eksploitasi terhadap aplikasi *web* DVWA.

III. HASIL DAN PEMBAHASAN

A. Reconnaissance

Reconnaissance adalah tahap persiapan yang penting yang dilakukan oleh penyerang sebelum mencuri informasi dari sebuah *server web*. Selama tahap ini, tujuan utamanya adalah untuk mengumpulkan informasi tentang *server* aplikasi *web* target dengan mengidentifikasi kerentanan yang dapat dieksploitasi. Berdasarkan kumpulan informasi tersebut menjadi sebuah panduan untuk melakukan eksploitasi terhadap target [3].

B. Spesifikasi Hardware dan Software

Spesifikasi *Hardware* dan *Software* yang digunakan untuk melakukan proses pengujian dan penelitian untuk melakukan eksploitasi terhadap aplikasi *web* DVWA dapat dilihat 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	20393MB RAM
	Hard Disk	120 GB SSD
	System Type	64-Bit
	Operating System	Linux Ubuntu 22.04 LTS
Spesifikasi Main OS	Processor	AMD Ryzen™ 7 Mobile Processors with Radeon™ Graphics
	Memory	16384MB RAM
	Hard Disk	500 GB SSD
	System Type	64-bit Operating System, x64-based processor
	Operating System	Windows 11 Home Single Language 64-bit (22H2, Build 22621)
Spesifikasi Virtual Machine	Processor	AMD Ryzen™ 7 Mobile Processors with Radeon™ Graphics
	Memory	4096 MB 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
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 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

a. Kali Linux

Kali Linux adalah sebuah distribusi Linux Debian sumber terbuka yang dirancang untuk keperluan forensik komputer atau pengujian penetrasi dan dalamnya terdapat berbagai perangkat lunak untuk melakukan pengujian [4].

2. Web Application

b. DVWA

DVWA adalah sebuah aplikasi *web* untuk melakukan pengujian kerentanan yang dikembangkan menggunakan bahasa PHP dan *database* MySQL. Tujuan utamanya adalah menyediakan lingkungan yang aman dan legal bagi para profesional keamanan untuk menguji keterampilan, serta membantu pengembang dalam memahami dengan lebih jelas cara mengamankan aplikasi *web* secara lebih aman [5]. Tingkat kerentanan sistem keamanan pada aplikasi *web* DVWA dapat diatur sesuai dengan kebutuhan pengujian.

3. Web Application Firewall

c. ModSecurity

ModSecurity adalah perangkat lunak yang umum digunakan untuk memantau, *logging* dan manajemen aplikasi *web* secara *realtime* [6]. ModSecurity merupakan salah satu *web application firewall* yang berfungsi untuk melindungi aplikasi *web* dari berbagai macam serangan 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*. Pada penelitian ini, ModSecurity digunakan sebagai filterisasi aplikasi *web* DVWA dari berbagai macam jenis eksploitasi pada saat pengujian eksploitasi yang akan dilakukan.

4. *Attack Tools*d. *Sqlmap*

SQLMap adalah aplikasi atau alat *open source* yang disertakan dengan Kali Linux [7]. SQLmap digunakan untuk pengujian penetrasi dan secara otomatis dapat mendeteksi serta melakukan eksploitasi. Pada proses implementasinya SQLmap membutuhkan alamat URL untuk melakukan permintaan terhadap target yang rentan dengan tujuan mendapatkan database pada aplikasi web yang pada akhirnya dapat memperoleh informasi tentang struktur basis data, termasuk nama basis data, konten tabel basis data, konten kolom basis data, hingga data yang tersedia di kolom basis data.

e. *Wfuzz*

Wfuzz merupakan salah satu alat untuk melakukan fuzzer yang dibangun dengan bahasa python dengan memiliki banyak komponen yang dapat dikembangkan [8]. Pada proses implementasinya wfuzz mengirimkan serangkaian permintaan HTTP dengan mengubah nilai *paramater* secara otomatis dan mengirimkan berbagai nilai *paramater* untuk mencoba menemukan celah kerentanan pada aplikasi web.

f. *Burpsuite*

Burpsuite merupakan *tool* yang digunakan pada penelitian ini untuk membantu mengidentifikasi, mengeksploitasi, dan mencari kerentanan pada aplikasi web DVWA. Penggunaan Burpsuite sebagai proxy dapat mencatat HTTP *request* yang dilakukan penyerang ketika mengakses DVWA. Dari HTTP *request* tersebut terdapat beberapa informasi seperti *cookie*, *request header*, URL yang dapat digunakan untuk melakukan eksploitasi.

g. *Firefox*

Firefox digunakan untuk melakukan perubahan *script* pada *website* yang akan dilakukan ketika pengujian keamanan dan juga untuk melakukan identifikasi lalu lintas HTTP *request* dan *post* pada *website* sehingga menghasilkan informasi yang dapat digunakan untuk melakukan eksploitasi terhadap aplikasi web DVWA.

5. *Vulnerability Scanning*h. *OWASP-ZAP*

OWASP-ZAP merupakan sebuah *tool* yang digunakan untuk melakukan *vulnerability scanning* pada aplikasi web. Penelitian kali ini menggunakan OWASP-ZAP untuk menemukan celah keamanan pada aplikasi web DVWA yang akan dimanfaatkan untuk melakukan eksploitasi.

C. *Scanning*

Vulnerability Scanning adalah tahap sebuah *web* akan diidentifikasi untuk diuji apakah memiliki kerentanan atau tidak, dan seberapa parah kerentanan tersebut. Jika ada kerentanan, pengujian akan menggunakan kerentanan tersebut untuk melanjutkan langkah pengujian selanjutnya. Hasil dari pemindaian pengujian berupa laporan yang berisi informasi mengenai celah kerentanan [9]. Pada tahap ini menampilkan informasi celah keamanan yang didapatkan hasil dari *scanning* menggunakan OWASP-ZAP untuk diidentifikasi mana saja celah yang rentan dari aplikasi *web*

DVWA serta dipilih untuk pengujian eksploitasi oleh penyerang.

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

Pada Tabel 3 merupakan hasil *report* yang didapatkan dari proses *scanning* pada aplikasi *web* DVWA. 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 kerentanan yang berhasil didapat, pada penelitian ini dibatasi jumlah serangan yang akan eksploitasi yaitu pada Tabel 4:

TABEL 4.

Pemilihan Eksploitasi Berdasarkan Hasil *Vulnerability Scanning* OWASP-ZAP

Descriptsi	Risk Level	CWE ID	WASC ID	Alert ID
Cross Site Scripting (Reflected)	High	79	8	Active (40012 - Cross Site Scripting (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)

D. *Analisis Perbandingan Metrik Time*

Pengukuran *time* yang dilakukan berdasarkan eksploitasi penyerangan mengacu pada berapa lama proses eksploitasi terhadap target aplikasi *web* DVWA sesuai dengan langkah langkah yang telah ditentukan berdasarkan eksperimen. Nilai dari pengukuran *time* bertipe detik (s). Berikut adalah tabel yang menunjukkan metrik *time* dari eksploitasi yang menunjukkan hasil dari data eksperimen aplikasi *web* dengan perlindungan WAF dan tanpa perlindungan WAF.

TABEL 5.

Analisis Perbandingan Metrik *Time* tanpa WAF

Number	Serangan	Time (s)		
		Real	User	Sys

1	Attack Tree SQL Injection	621,34s	1.112,02s	368,61s
2	Attack Tree Cross-Site Scripting (Reflected)	52,88s	20,07s	7,92s
3	Attack Tree Command Injection	74,44s	9,94s	4,87s
4	Attack Tree Cross Site Request Foreign	206,20s	61,90s	23,00s
5	Attack Tree Brute Force	206,58s	125,20s	40,26s

TABEL 6.
Analisis Perbandingan Metrik Time dengan WAF

Number	Serangan	Time (s)		
		Real	User	Sys
1	Attack Tree SQL Injection	48,15s	25,65s	9,13s
2	Attack Tree Cross-Site Scripting (Reflected)	45,50s	12,31s	5,64s
3	Attack Tree Command Injection	91,70s	9,54s	4,91s
4	Attack Tree Cross Site Request Foreign	109,27s	55,71s	20,26s
5	Attack Tree Brute Force	161,35s	61,67s	14,45s

Pengukuran metrik *time* dicatat 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 menggunakan rumus sebagai berikut:

$$\sum_{i=1}^n t(A) = r_1 + r_2 + \dots + r_n \dots (i)$$

Dengan:

$t(A)$ = Attack time (detik)

n = batas atas

i = indeks penjumlahan

r = real time

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

TABEL 7.
Perbandingan Metrik Time Antar Eksploitasi tanpa WAF

Serangan	Time Metrik (s)
Attack tree SQL Injection	621,34s
Attack tree XSS (Reflected)	52,88s
Attack tree Command injection	74,44s
Attack tree CSRF	206,20s

Attack tree Brute force	206,58
-------------------------	--------

TABEL 8.
Perbandingan Metrik Time Antar Eksploitasi dengan WAF

Serangan	Metrik time(s)
Attack tree SQL Injection	48,15s
Attack tree XSS (Reflected)	45,50s
Attack tree Command injection	91,70s
Attack tree CSRF	109,27
Attack tree Brute force	161,35s

E. Analisis Perbandingan Metrik Skill Level

Pengukuran dicatat menggunakan aspek *skill level* yaitu berdasarkan penentuan hasil analisis tingkat kemampuan suatu tahapan eksploitasi. Pengukuran dilakukan perhitungan menggunakan rumus berikut:

$$\sum_{i=1}^n S(A) = x_1 + x_2 + \dots + x_n \dots (i)$$

Dengan:

$S(A)$ = Skill Level Attack

x = Skill Level Step Eksploitasi

Berikut merupakan tabel yang menunjukkan nilai *skill level* suatu tahapan eksploitasi terhadap aplikasi berbasis web dengan kondisi tidak terlindungi dan dengan terlindungi oleh WAF:

TABEL 9.
Perbandingan Metrik Skill Level Eksploitasi tanpa WAF

Serangan	Total Skill level
Attack tree SQL Injection	13
Attack tree XSS (Reflected)	4
Attack tree Command injection	8
Attack tree CSRF	7
Attack tree Brute force	8

TABEL 10.
Perbandingan Metrik Skill Level Eksploitasi dengan WAF

Serangan	Total Skill level
Attack tree SQL Injection	4
Attack tree XSS (Reflected)	4
Attack tree Command injection	8
Attack tree CSRF	7
Attack tree Brute force	8

F. Analisis Perbandingan Attack Tree Berdasarkan Metrik Time dan Skill Level

Dalam menentukan perbandingan terhadap kedua metrik yaitu *time* dan *skill level*, diperlukan suatu rumus untuk melakukan perhitungan perbandingan yang dapat menghasilkan sebuah nilai atau angka yang menjadi acuan untuk pemeringkatan suatu pengujian eksploitasi berdasarkan metrik *time* dan *skill level*. Melalui perhitungan dengan rumus akan menghasilkan sebuah kalkulasi yang dinamakan sebagai “*time skill level score*” yang merepresentasikan nilai skor perbandingan dari kedua metrik tersebut. Nilai “*time skill level score*” dapat digunakan untuk mencari skor perbandingan dengan

menggunakan metrik *time* dan *skill level*. Selanjutnya, akan dilakukan pemeringkatan berdasarkan serangan mana yang memiliki skor yang paling rendah hingga tertinggi. Berikut merupakan rumus perhitungan untuk perbandingan antar metrik:

$$x(A) = (r_1 \cdot s_1) + (r_2 \cdot s_2) + \dots (r_n \cdot s_n)$$

Dengan:

$x(A)$ = *time skill level score* pada eksploitasi

r = *real time* yang dibutuhkan untuk eksploitasi

s = *skill level* step eksploitasi untuk eksploitasi

Setelah dilakukan perhitungan dengan menggunakan rumus tersebut, diperoleh suatu kalkulasi data yang dinamakan sebagai *Time Skill level Score*. Selanjutnya akan dilakukan pemeringkatan berdasarkan serangan mana yang memiliki skor yang paling rendah hingga tertinggi. Berikut merupakan tabel yang menampilkan hasil pemeringkatan metrik *time* dan *skill level* pada pengujian eksploitasi tanpa WAF dan dengan perlindungan WAF.

TABEL 11.

Pengurutan *Time* dan *Skill Level* Metrik pada Pengujian Eksploitasi tanpa Perlindungan WAF

Rank	Serangan	Time Skill Level Score
1	Attack tree XSS (Reflected)	131,92
2	Attack tree Command injection	159,39
3	Attack tree CSRF	465,18
3	Attack tree Brute force	488,97
4	Attack tree SQL Injection	1727,56

Pada Tabel 11 menunjukkan bahwa pada urutan pertama eksploitasi “XSS (Reflected)” menjadi urutan pertama dengan skor 131,92 dan eksploitasi “SQL Injection” menjadi urutan terakhir dengan skor 1727,56.

TABEL 12.

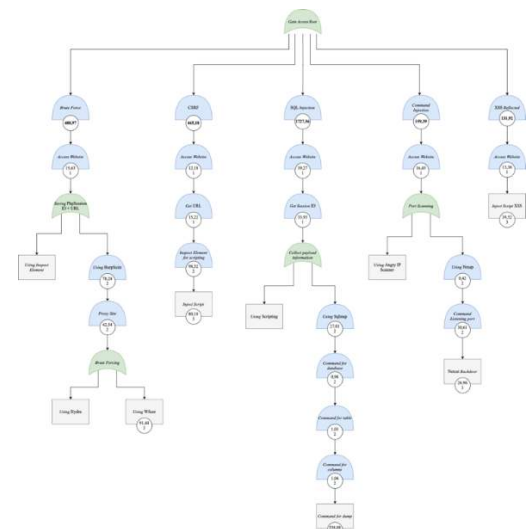
Pengurutan *Time* dan *Skill Level* Metrik pada Pengujian Eksploitasi dengan WAF

Rank	Serangan	Time Skill Level Score
1	Attack tree SQL Injection	54
2	Attack tree XSS (Reflected)	96,12
3	Attack tree Command injection	198,7
4	Attack tree CSRF	234,68
5	Attack tree Brute force	319,51

Pada Tabel 12 menunjukkan bahwa pada urutan pertama eksploitasi “SQL Injection” menjadi urutan pertama dengan skor 54 dan eksploitasi “Brute force” menjadi urutan terakhir dengan skor 319,51.

Setelah mendapatkan perbandingan dari kedua metrik yaitu metrik *time* dan *skill level* dengan hasil *time skill level score* kemudian untuk hasil akhir akan dibuatkan secara visual melalui *attack tree* yang menampilkan *time skill level score* serta *time* dan *skill level* pada setiap *node attack tree* yang akan merepresentasikan tahapan dari eksploitasi.

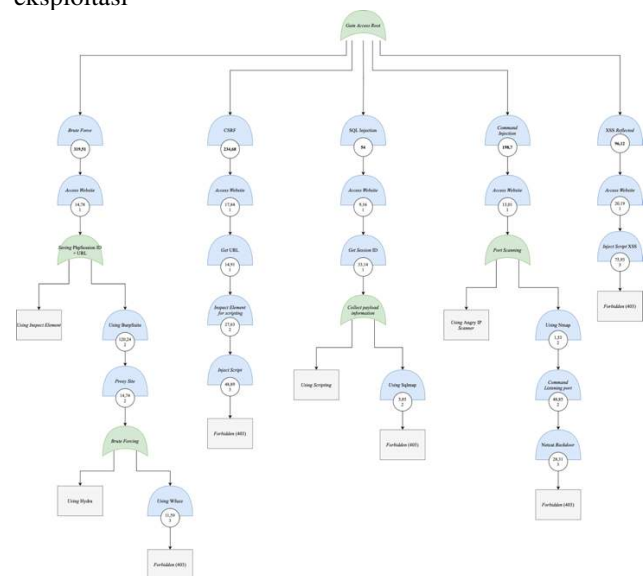
Berikut merupakan *attack tree* dengan metrik *time* dan *skill level* tanpa WAF:



GAMBAR 2.

Diagram Attack Tree Metrik Time dan Skill Level tanpa WAF

Pada Gambar 2 menjelaskan tentang diagram *attack tree* dari semua serangan yang dibuat berdasarkan eksploitasi. Semua serangan bertujuan untuk mendapatkan akses *root* secara ilegal. Pada *diagram* ini memuat lima serangan yaitu *Brute force*, *CSRF*, *SQL Injection*, *Command injection*, dan *XSS Reflected* yang berisikan masing masing langkah dan pengukuran metrik *time skill level*. Ukuran *time* pada masing masing langkah pada *attack tree* berupa akumulasi waktu dari step mulai hingga selesai. Ukuran *skill level* pada masing masing langkah pada *attack tree* yaitu nilai tingkat kemampuan dari setiap urutan langkah pada proses eksploitasi



GAMBAR 3.

Diagram Attack Tree Metrik Time dan Skill Level dengan WAF

Pada Gambar 3 menjelaskan tentang diagram *attack tree* dari semua serangan yang dibuat berdasarkan eksploitasi. Semua serangan bertujuan untuk mendapatkan akses *root*

secara ilegal. Pada *diagram* ini memuat lima serangan yaitu *Brute force*, *CSRF*, *SQL Injection*, *Command injection*, dan *XSS Reflected* yang berisikan masing masing langkah dan pengukuran metrik *time skill level*. Ukuran *time* pada masing masing langkah pada *attack tree* berupa akumulasi waktu dari step mulai hingga selesai. Ukuran *skill level* pada masing masing langkah pada *attack tree* yaitu nilai tingkat kemampuan dari setiap urutan langkah pada proses eksploitasi. Sehingga skor untuk pemeringkatan dapat dilihat pada nama serangan eksploitasi. Bentuk perlindungan dari WAF adalah dengan memutuskan koneksi dari penyerang dan memberikan halaman yang menampilkan pesan “403 Forbidden”.

IV. KESIMPULAN

Berdasarkan analisa pada bagian sebelumnya, peneitian ini menghasilkan kesimpulan sebagai berikut:

1. *Activity diagram* dan *data flow diagram* dibuat berdasarkan hasil tahapan eksploitasi untuk menggambarkan *attack tree*.
2. *Attack tree* dapat disusun berdasarkan karakter menggunakan relasi metrik *time* dan *skill level*.
Metrik *time* dan *skill level* dapat digunakan untuk pemeringkatan berbagai *attack tree*. Peringkat tertinggi tanpa WAF adalah *attack tree XSS (Reflected)* dengan skor 131,92. *Attack tree SQL Injection* menempati urutan terakhir dengan skor 1727,56. Pemeringkatan dapat dipengaruhi oleh peranan WAF dengan memutus *node* tahapan *attack tree*. Dengan skor tertinggi *attack tree SQL Injection* 54. *Attack tree brute force* menempati urutan terakhir dengan skor 319,51.

REFERENSI

- [1]. Dhiatama Ayunda, K., Widjajarto, A., & Budiono, A., “*Implementation and Analysis ModSecurity on Web-Based Application with OWASP Standards*”, Jurnal Teknik Informatika dan Sistem Informasi, Vol.8, No.3, pp. 1638–1650, September 2021.
- [2]. Kuipers, L. (2020). *Analysis of Attack Trees: fast algorithms for subclasses*.
- [3]. Odun-Ayo *et al.*, ‘*Evaluating Common Reconnaissance Tools and Techniques for Information Gathering*’, *Journal of Computer Science*, vol. 18, no. 2, pp. 103–115, 2022, doi: 10.3844/jcssp.2022.103.115.
- [4]. D. Bhatt, ‘*Modern Day Penetration Testing Distribution Open Source Platform-Kali Linux-Study Paper*’, *INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH*, vol. 7, 2018, [Online]. Available: www.ijstr.org
- [5]. S. Tyagi and K. Kumar, ‘*Evaluation of static web vulnerability analysis tools*’, in *PDGC 2018 - 2018 5th International Conference on Parallel, Distributed and Grid Computing*, Institute of Electrical and Electronics Engineers Inc., Dec. 2018, pp. 1–6. doi: 10.1109/PDGC.2018.8745996.
- [6]. 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*”.
- [7]. S. Lika, R. D. P. Halim, and I. Verdian, ‘*ANALISA SERANGAN SQL INJEKSI MENGGUNAKAN SQLMAP*’, *POSITIF: Jurnal Sistem dan Teknologi Informasi*, vol. 4, no. 2, p. 88, Nov. 2018, doi: 10.31961/positif.v4i2.610.
- [8]. 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)*.
- [9]. E. Z. Darajat, E. Sedyono, and I. Sembiring, ‘*Vulnerability Assessment Website E-Government dengan NIST SP 800-115 dan OWASP Menggunakan Web Vulnerability Scanner*’, *JURNAL SISTEM INFORMASI BISNIS*, vol. 12, no. 1, pp. 36–44, Sep. 2022, doi: 10.21456/vol12iss1pp36-44.