KESATRIA: Jurnal Penerapan Sistem Informasi (Komputer & Manajemen)
Terakreditasi Nomor 204/E/KPT/2022 | Vol. 4, No. 4, Oktober (2023), pp. 841-851

# Implementasi dan Analisis Attack Tree pada Aplikasi DVWA Berdasar Metrik Time dan Cost

*Alifurfan Wiradwipa Pranowo[1], Adityas Widjajarto[2], Muhammad Fathinuddin[3]*
*[1,2,3]Universitas Telkom, Indonesia*
*E-mail: alifurfan@student.telkomuniversity.ac.id, adtwjrt@telkomuniversity.ac.id, muhammadfathinuddin@telkomunivertity.ac.id*

***Abstract***

*Exploitation against web applications can be formulated into an attack tree. This research aims to explore the relationship between the attack tree and the exploitation characteristics based on time and cost metrics. The study involves conducting exploitation experiments on the DVWA platform. The exploitation stages are utilized to construct the attack tree, which is then organized based on two conditions: with Web Application Firewall (WAF) and without WAF. The attack tree is composed of five types of exploitation, namely SQL Injection, XSS (Reflected), Command Injection, CSRF, and Brute Force. The analysis results without WAF indicate that the XSS (Reflected) attack tree occupies the top position with a score of 53.69, while the SQL Injection attack tree ranks last with a score of 682.49. On the other hand, with WAF, the XSS (Reflected) attack tree remains at the top with a score of 61.11, and the SQL Injection attack tree still occupies the last position, but with a lower score of 207.22. Consequently, this relationship can be utilized to categorize attack trees based on time and cost metrics. Future research opportunities may involve measuring subsystem processes of the system.*

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

***Abstrak***

*Eksploitasi terhadap aplikasi web dapat dirumuskan menjadi attack tree. Penelitian ini bertujuan untuk mengetahui relasi attack tree dengan karakter eksploitasi berdasarkan metrik time dan cost. Penelitian berdasarkan percobaan eksploitasi pada platform DVWA. Tahapan eksploitasi digunakan untuk meyusun attack tree. Penyusunan attack tree berdasarkan kondisi WAF dan tanpa WAF. Attack tree disusun berdasarkan lima eksploitasi yaitu SQL Injection, XSS (Reflected), Command Injection, CSRF, dan Brute Force. Hasil analisis tanpa WAF menghasilkan XSS(Reflected) attack tree menempati posisi pertama dengan skor 53,69. SQL Injection attack tree menempati urutan terakhir dengan skor 682,49. Sedangkan dengan WAF menghasilkan XSS(Reflected) attack tree menempati posisi pertama dengan skor 61,11. SQL Injection attack tree menempati urutan terakhir dengan skor 207,22. Dengan demikian relasi ini dapat digunakan untuk melakukan pengkategorian antar attack tree berdasarkan metrik time dan cost. Peluang penelitian selanjutnya dapat berupa pengukuran sub proses sistem.*

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

## 1. Pendahuluan

Kebutuhan dan penggunaan akan teknologi jaringan komputer semakin meningkat. Selain sebagai media penyedia informasi, melalui internet pula komunikasi menjadi bagian terbesar dan pesat pertumbuhannya serta menembus berbagai batas negara. Akan tetapi dampak negatif pun tidak bisa dihindari. Salah satunya adalah dapat menyebabkan kemungkinan munculnya kejahatan yang disebut dengan *cybercrime*. Keamanan siber menjadi perhatian belakangan ini sehingga ada perlunya sedikit dari ribuan kemungkinan penyerangan siber dapat dianalisa dan dikaji lebih lanjut untuk menangkal dan membuat

ISSN: 2720-992X

pertahanan pada perangkat komputer, salah satunya dengan penerapan *Web Aplication Firewall*.

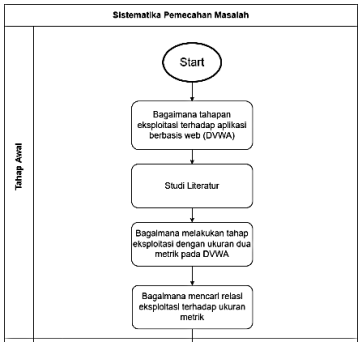
WAF merupakan sebuah *firewall* untuk aplikasi *web* yang berfungsi untuk filterisasi lalu lintas jaringan.[1] Filterisasi paket data dari lalu lintas jaringan yang ditemukan, dapat dilakukan blokir paket-paket data yang di curigai lalu dilakukan pencatatan aktifitas tersebut sehingga terlihat data untuk dianalisis lebih lanjut. Salah satu contoh dari WAF adalah ModSecurity. Untuk mendapatkan keamanan yang lebih lanjut, dilakukan analisis terhadap pengujian eksploitasi serangan yang menggunakan perlindungan dari WAF maupun non WAF agar terlihat hasil perbandingan dari pengujian tersebut.

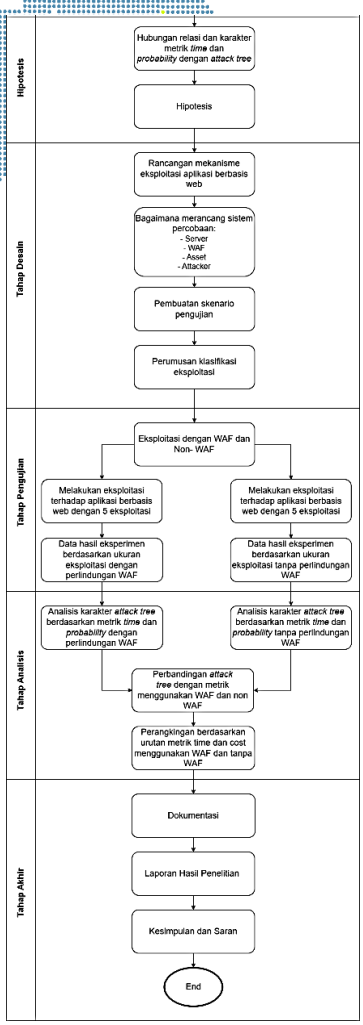
Pada penelitian ini menggunakan *Web Application Firewall* untuk melakukan perlindungan terhadap objek yang dipakai dalam studi kasus penelitian. Dasar kinerja dari WAF yaitu dapat melindungi objek dari serangan seperti *Brute Force*, *SQL Injection*, *XSS*, dan serangan-serangan lainnya. Pengujian pada penelitian kali ini dilakukan berdasarkan eksploitasi dan menggunakan standar dari OWASP TOP TEN yang menjadi acuan kerangka penyerangan pada objek. Hasil dari eksploitasi lantas diolah menjadi sebuah kerangka penyerangan yang disebut dengan *attack tree*. *Attack tree* merupakan metodologi yang dapat menjelaskan keamanan sebuah sistem dengan berisikan berbagai kemungkinan dari serangan eksploitasi penyerang untuk dilakukan analisis lebih lanjut sebagai pencegahan keamanan sistem.

Pada tugas akhir ini, dilakukan implementasi dari peran kinerja WAF terhadap pengujian eksploitasi untuk diolah dan dianalisis yang menjadi hasil akhir berupa catatan pengukuran dan kumpulan data informasi yang diolah menjadi metrik metrik dan diagram penyerangan eksploitasi. Metode serangan eksploitasi yang dipakai dalam pengujian diambil berdasarkan hasil vulnerability scanning dan pengujian dilakukan dengan dua kondisi yaitu pada saat *web* aplikasi dalam perlindungan WAF maupun tidak. Pada analisis, dilakukan perbandingan hasil data pengujian berdasarkan metrik metrik yang diukur pada proses pengujian WAF. Hasil analisis yang didapat bertujuan untuk mengetahui karakter eksploitasi berdasarkan metrik tertentu menggunakan *attack tree*.

## 2. Metodologi Penelitian

Sistematika penelitian disajikan sebagai alur yang tersusun dan menjadi panduan untuk memecahkan permasalahan yang muncul ketika penelitian. Terdapat enam tahapan utama metodologi yang harus dilakukan yaitu tahap awal, hipotesa, tahap desain, tahap pengujian, tahap analisis dan tahap akhir. Berikut adalah diagram yang menjelaskan sistematika penyelesaian masalah.






**Gambar 1.** Sistematika Penyelesaian

1. Tahap Awal (Perumusan Masalah)

Tahap awal penelitian dimulai dengan melakukan identifikasi tentang bagaimana tahapan eksploitasi terhadap objek DVWA yaitu aplikasi berbasis *web*. Dilanjutkan dengan melakukan riset terkait dengan studi literatur. Studi literatur juga dilakukan untuk memastikan bahwa masalah yang diangkat memiliki relevansi dan memungkinkan untuk dilakukan dan memperdalam teori mengenai tahapan eksploitasi terhadap objek penelitian. Pengujian eksploitasi dilaksanakansetelah tahapan studi literatur dilaksanakan. Pengujian eksploitasi dilaksanakan berkaitan dengan dua metrik yang digunakan pada penelitian ini yaitu metrik *time* dan *cost*. Selanjutnya masuk ke fase analisa terhadap relasi yang berkaitan antar metrik *time* dan *cost* berdasarkan eksploitasi.

2. Tahap Hipotesis

Setelah tahap awal terlaksana, dilanjutkan ke tahap selanjutnya yaitu tahap hipotesis. Pada tahap ini melakukan hipotesis yang menghasilkan praduga terhadap hipotesis yang berkaitan dengan relasi dan karakter metrik yaitu *time* dan *cost* dengan *attack tree*.

3. Tahap Desain

Pada tahap ini akan berfokus kepada tahap persiapan desain yang menyangkut perancangan eksploitasi. Skenario pengujian dibuat setelah instalasi perangkat lunak pada mesin virtual dan server yang terdiri dari:

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

Perancangan skenario pengujian dimulai dengan melakukan pemindaian terhadap objek aplikasi berbasis *web*. Hasil dari pemindaian digunakan untuk mengklasifikasikan jenis eksploitasi.

4. Tahap Pengujian

Pada tahap ini akan dilaksanakan tahapan eksploitasi dengan 5 jenis metode eksploitasi terpilih yang memiliki dua kondisi pengujian eksploitasi yaitu sebagai berikut:

a) Eksploitasi dengan kondisi WAF non-aktif.
b) Eksploitasi dengan kondisi WAF aktif.

Setelah pengujian eksploitasi dilaksanakan dengan kedua kondisi selesai, akan menghasilkan sebuah data eksperimen berdasarkan ukuran pengujian eksploitasi terhadap objek DVWA. Data yang dihasilkan antara lain:

a) Data hasil eksperimen ukuran eksploitasi dengan kondisi WAF non-aktif.
b) Data hasil eksperimen eksploitasi ukuran dengan kondisi WAF aktif.

Setelah ukuran dengan dua kondisi diketahui maka dapat dilanjutkan ke tahapan selanjutnya yaitu tahap analisis.

5. Tahap Analisis

Pada tahap ini akan dilaksanakan analisis karakter yang digambarkan dengan diagram *attack tree*. Karakteristik *attack tree* dianalisis berdasarkan dengan data hasil yang berisikan metrik terkait dengan *time* dan *cost* dengan dua kondisi yaitu:

a) Analisis eksploitasi dengan kondisi WAF non-aktif.
b) Analisis eksploitasi dengan kondisi WAF aktif.

Analisis juga dilakukan untuk mengukur eksploitasi yang menghasilkan data metrik *time* dan *cost*. Hasil analisis ini akan menunjukkan karakteristik *attack tree* yang berkaitan dengan metrik *time* dan *cost*. Selanjutnya, akan dilakukan perbandingan yang digunakan untuk menyusun pola *attack tree* berdasarkan kondisi aplikasi berbasis *web* dengan perlindungan dan tanpa perlindungan WAF. Setelah hasil data terbandingkan, data akan diolah dengan melakukan pengkategorian berdasarkan urutan metrik *time* dan *cost*.

6. Tahap Akhir

Tahap ini adalah tahap terakhir dari penelitian. Penarikan kesimpulan terkait dengan hasil pengujian eksploitasi yang menghasilkan data akhir untuk analisis mengenai karakter *attack tree* berdasarkan metrik time dan cost serta pemberian saran akan dituliskan pada laporan hasil penelitian.

## 3. Hasil Dan Pembahasan

### 3.1. *Reconnaissance*

*Reconnaissance* merupakan tahap persiapan yang penting yang dilakukan oleh penyerang sebelum mencuri informasi dari sebuah *server web*. Selama tahap ini, tujuan utamanya adalah untuk mengumpulkan sebanyak mungkin informasi tentang server *web* target. Teknik yang digunakan dalam *reconnaissance* melibatkan pemindaian jaringan dari perspektif internal dan eksternal, kegiatan ini dilakukan tanpa mendapatkan izin dari pemilik *server*.

### 3.2. Spesifikasi Hardware dan Software

Perangkat pada pengujian dibutuhkan untuk memberikan sumber daya pada pengujian dan sebagai alat proses dari segi system. Perangkat pada pengujian meliputi perangkat keras dan perangkat lunak yang digunakan selama proses pengujian. Tabel 1 adalah tabel yang menjelaskan spesifikasi perangkat keras selama proses pengujian dan penelitian. Berikut adalah rincian perangkat keras yang digunakan:

**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 |
| *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 |

Menurut Tabel 2 disebutkan beberapa spesifikasi yang digunakan pada penelitian dan pengujian, pada bagian ini akan dijelaskan tentang fungsi dari setiap perangkat lunak yang digunakan yaitu sebagai berikut:

1. *Operating System*

Kali Linux adalah sebuah distro Linux yang dirancang khusus untuk keperluan keamanan komputer dan *penetration testing*.[2] Kali linux menyediakan alat yang lengkap dan terintegrasi dalam satu sistem operasi yang bertujuan untuk menjalankan aktivitas pengujian keamanan, seperti pemindaian jaringan, analisis kerentanan, forensik komputer, dan banyak lagi.

2. *Web Application*

DVWA adalah *web application* yang sengaja dibuat rentan dengan tujuan untuk memberikan kemudahan dalam simulasi pengujian keamanan pada *web application*. Fungsi utama DVWA adalah untuk memberikan platform yang terbuka dan legal bagi para penguji keamanan, pengembang, dan peneliti untuk mempelajari serta melatih kemampuan mereka dalam mengidentifikasi kerentanan dan melaksanakan penetration testing pada aplikasi *web*. [3]

3. *Web Application Firewall*

ModSecurity berfungsi untuk melindungi aplikasi *web* dari serangan dan ancaman keamanan dengan mendeteksi dan mencegah serangan terhadap aplikasi *web* menggunakan penerapan aturan keamanan khusus yang disebut dengan SecRule. ModSecurity menggunakan *rule* yang berfungsi sebagai filter di antara aplikasi *web* dan pengguna.[4] Transmisi data pada lalu lintas HTTP dianalisis dan dapat mengidentifikasi

serangan serta memblokir serangan tersebut sebagai aktifitas terlarang dan dilakukan pencatatan pada *log* aktifitas *firewall*.

4. *Attack Tools*

SQLMap adalah sebuah *tool* untuk pengujian penetrasi *database* yang digunakan untuk mendeteksi dan mengeksploitasi kerentanan pada aplikasi *web* yang menggunakan *database* SQL. Hasil ekspoitasi dan ekstraksi dari Sqlmap dapat menghasilkan sebuah informasi sensitif dari database, tabel, kolom, dan data yang tersedia.

Wfuzz adalah sebuah *tool* untuk melakukan serangan fuzzing pada aplikasi *web*. Dalam prosesnya Fuzzing mengirimkan serangkaian permintaan HTTP dengan mengubah nilai parameter secara otomatis lalu WFuzz mengirimkan permintaan dengan variasi nilai parameter yang bervariasi untuk mencoba menemukan celah kerentanan.

Burp Suite adalah sebuah *tool platform* lengkap untuk pengujian keamanan aplikasi *web*. Burp Suite berfungsi untuk membantu dalam mengidentifikasi, mengeksploitasi, dan mencari kerentanan pada aplikasi *web*. Burp Suite memiliki beberapa fitur yang terdiri dari *web proxy*, *vulnerability scanning*, *spider*, *repeater*, *intruder*, *sequencer*, dan lain lain.

Firefox adalah sebuah *web browser* yang dikembangkan oleh Mozilla Corporation. Firefox berfungsi untuk memuat dan menampilkan halaman *web* secara visual kepada pengguna. Firefox dapat berfungsi juga untuk melakukan perubahan pada *script web*site untuk melancarkan rekayasa penyerangan.

5. *Vulnerability Scanning*

OWASP-ZAP atau kepanjangan dari Open *Web* Application Security Project - Zed Attack Proxy adalah sebuah *tool* otomatis untuk pengujian penetrasi yang dirancang untuk mengidentifikasi dan mengeksploitasi kerentanan keamanan pada aplikasi *web*.

### 3.3. *Scanning*

*Vulnerability scanning* adalah proses identifikasi dan evaluasi kerentanan keamanan pada suatu sistem komputer atau jaringan.[5] Pada tahap ini akan terlihat celah keamanan yang didapatkan hasil dari scanning menggunakan OWASP-ZAP untuk diidentifikasi mana saja celah yang rentan dari *server* target lalu dimanfaatkan untuk melakukan 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 |

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 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 | *High* | 78 | 31 | Active (90020 - Remote OS |

| Deskripsi | *Risk Level* | CWE ID | WASC ID | Alert ID |
|---|---|---|---|---|
| Injection | | | | 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*

Metrik *time* adalah pengukuran untuk melakukan observasi, pengukuran, dan pencatatan setiap interval waktu yang terjadi pada proses pengujian.[6] 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 |
| 1 | *Attack Tree SQL Injection* | 682,49 | 619,55 | 96 |
| 2 | *Attack Tree* XSS (*Reflected*) | 53,69 | 6,64 | 2,2 |
| 3 | *Attack Tree Command Injection* | 105,29 | 6,42 | 2,99 |
| 4 | *Attack Tree* CSRF | 198,03 | 22,19 | 4,67 |
| 5 | *Attack Tree Brute Force* | 221,90 | 50,40 | 6,09 |

**Tabel 6.** Analisis Perbandingan Metrik *Time* dengan WAF

| Number | Serangan | Time (s) | | |
|---|---|---|---|---|
| | | Real | User | Sys |
| 1 | *Attack Tree SQL Injection* | 83 | 15,09 | 5,01 |
| 2 | *Attack Tree* XSS (*Reflected*) | 61,11 | 7,48 | 2,02 |
| 3 | *Attack Tree Command Injection* | 113,04 | 12,23 | 3,38 |
| 4 | *Attack Tree* CSRF | 207,22 | 28,74 | 6,88 |
| 5 | *Attack Tree Brute Force* | 179,51 | 48,39 | 11,9 |

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 dilakukan perhitungan terhadap pengukuran metrik *time*, maka di-identifikasikan metrik *time* dari setiap serangan tanpa perlindungan WAF:

**Tabel 7.** Perbandingan Metrik Time antar Eksploitasi tanpa WAF

| Number | Serangan | Time Metric (s) |
|---|---|---|
| 1 | Attack Tree SQL Injection | 682,49 |
| 2 | Attack Tree XSS (Reflected) | 53,69 |

| Number | Serangan | Time Metric (s) |
|---|---|---|
| 3 | Attack Tree Command Injection | 105,29 |
| 4 | Attack Tree CSRF | 198,03 |
| 5 | Attack Tree Brute Force | 221,90 |

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

| Number | Serangan | Time Metric (s) |
|---|---|---|
| 1 | *Attack Tree SQL Injection* | 83 |
| 2 | *Attack Tree* XSS *(Reflected)* | 61,11 |
| 3 | *Attack Tree Command Injection* | 113,04 |
| 4 | *Attack Tree* CSRF | 207,22 |
| 5 | *Attack Tree Brute Force* | 179,51 |

### 3.5. Analisis Perbandingan Metrik *Cost*

Pengukuran nilai yang dilakukan berdasarkan penyerangan yang mengacu kepada eksploitasi memiliki nilai yang dihitung berdasarkan seberapa banyak proses yang dibutuhkan pada proses pengujian. Pengukuran dicatat menggunakan aspek *cost* yaitu berdasarkan langkah langkah pengujian pada serangan. pengukuran dilakukan perhitungan menggunakan rumus berikut:

$$\sum_{i=1}^{n} c(A) = s_1 + s_2 + \cdots + s_n \quad \ldots\ldots (i) \qquad (2)$$

Dengan:

$c(A) = attack\ cost(step)$

$n =$ batas atas

$i =$ indeks penjumlahan

$s = step$ serangan

Berikut merupakan tabel yang menunjukkan perhitungan *cost* lagkah suatu tahapan eksploitasi terhadap aplikasi berbasis web dengan kondisi tidak terlindungi dan dengan terlindungi oleh WAF:

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

| Serangan | Cost (Step) |
|---|---|
| *Attack Tree SQL Injection* | 2 |
| *Attack Tree* XSS *(Reflected)* | 4 |
| *Attack Tree Command Injection* | 4 |
| *Attack Tree* CSRF | 4 |
| *Attack Tree Brute Force* | 7 |

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

| Serangan | Cost (Step) |
|---|---|
| *Attack Tree SQL Injection* | 2 |
| *Attack Tree* XSS *(Reflected)* | 3 |
| *Attack Tree Command Injection* | 4 |
| *Attack Tree* CSRF | 4 |
| *Attack Tree Brute Force* | 4 |

### 3.6. Analisis Perbandingan *Attack Tree* Berdasarkan Metrik *Time* dan *Cost*

Metrik *cost* adalah sebuah metode pengukuran yang diperoleh dengan menghitung jumlah langkah yang digunakan dalam suatu proses.[6] Pengukuran metrik *time* dicatat menggunakan tiga aspek *time* yaitu *: Real, User, dan Time*. Pada penelitian ini hanya diambil aspek *real* karena sudah mewakili dari dua aspek lainnya. Sedangkan pengukuran metrik *cost* dicatat menggunakan aspek *cost* yaitu berdasarkan langkah langkah pengujian

pada serangan. Perlunya dilakukan perhitungan untuk perbandingan antar metrik menggunakan rumus berikut:

$$\sum_{i=1}^{n} \mathrm{Z}(A) = (r_1.s_1) + (r_2.s_2), + \cdots (r_n.s_n) \quad (3)$$

Dengan:
$Z(A)$ = *Time Cost Score* pada eksploitasi
$r$ = *real time* yang dibutuhkan pada penyerangan
$s$ = *step* yang diperlukan pada penyerangan

Melalui perhitungan dengan rumus, menghasilkan sebuah kalkulasi yang dinamakan sebagai "time cost score" yang merepresentasikan nilai skor perbandingan dari kedua metrik tersebut. Nilai "time cost score" dapat digunakan untuk melakukan pengkategorian yang diurut berdasarkan serangan mana yang memiliki skor yang paling rendah hingga paling skor tinggi. Berikut adalah tabel hasil pengurutan berdasarkan pengkategorian dengan "time cost score":

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

| ***Rank*** | **Serangan** | ***Time Cost Score*** |
|---|---|---|
| 1 | *Attack Tree* XSS (Reflected) | **53,69** |
| 2 | *Attack Tree* Command Injection | **105,29** |
| 3 | *Attack Tree* CSRF | **198,03** |
| 4 | *Attack Tree* Brute Force | **223,17** |
| 5 | *Attack Tree* SQL Injection | **682,49** |

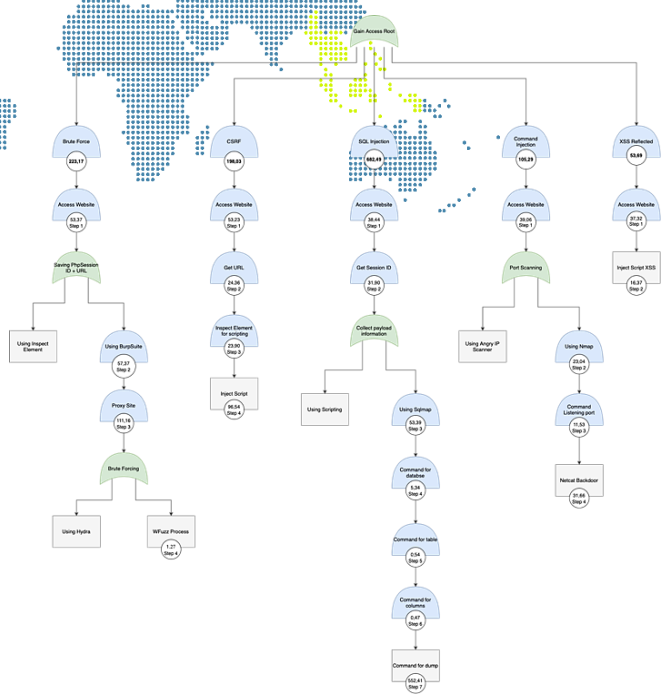
Tabel 11 menunjukkan bahwa pada urutan pertama eksploitasi "XSS (Reflected)" menjadi urutan pertama dengan skor 53,69 dan eksploitasi "SQL Injection" menjadi urutan terakhir dengan skor 682,49.
Pengukuran pada pengujian eksploitasi juga dilaksanakan terhadap aplikasi *web* dengan perlindungan WAF yaitu sebagai berikut:

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

| **Rank** | **Serangan** | ***Time Cost Score*** |
|---|---|---|
| 1 | *Attack Tree* XSS (Reflected) | **61,11** |
| 2 | *Attack Tree* SQL Injection | **83** |
| 3 | *Attack Tree* Command Injection | **113,04** |
| 4 | *Attack Tree* Brute Force | **179,51** |
| 5 | *Attack Tree* CSRF | **207,22** |

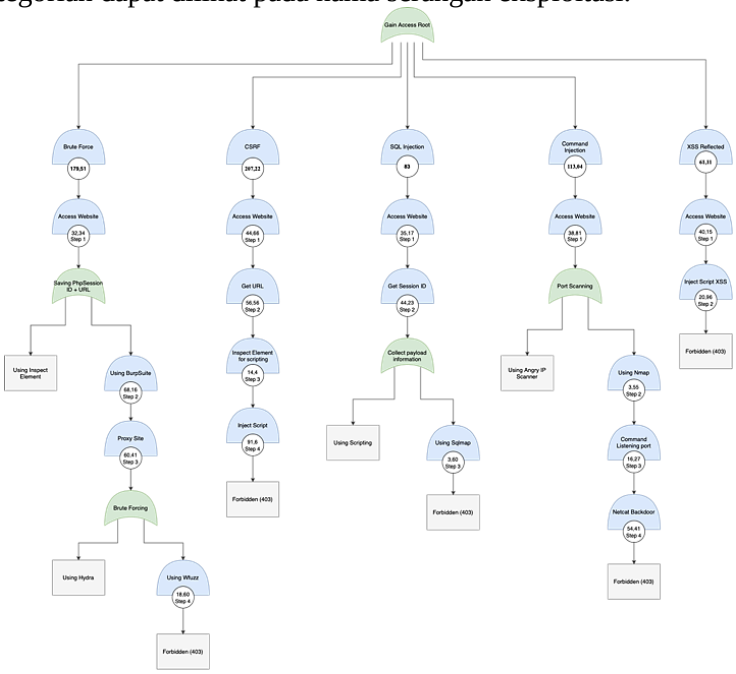
Tabel 12 menunjukkan bahwa pada urutan pertama eksploitasi "XSS (Reflected)" menjadi urutan pertama dengan skor 61,11 dan eksploitasi "CSRF" menjadi urutan terakhir dengan skor 207,22.

Setelah mendapatkan urutan *ranking* pada keseluruhan eksploitasi, kemudian hasil akhir dibuatkan visualisasi *attack tree. Attack tree adalah sebuah metodologi yang menjelaskan keamanan sistem yang dilandasi oleh kemungkinan dari macam-macam serangan. Struktur pada pohon menggambarkan serangan yang dilakukan kepada sebuah sistem.[7] Attack tree* yang berisikan *time cost score* serta *time* dan *cost* pada setiap *node attack tree* yang merepresentasikan tahapan pada eksploitasi. Berikut merupakan *attack tree* dengan metrik *time* dan *cost* dengan kondisi tanpa perlindungan dan dengan perlindungan WAF:




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

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 cost* dari hulu ke hilir tanpa menggunakan perlindungan dari WAF. Ukuran time pada masing masing langkah pada *attack tree* berupa akumulasi waktu dari step mulai hingga selesai. Ukuran cost pada masing masing langkah pada *attack tree* berupa keterangan urutan langkah pada proses penyerangan. Sehingga skor untuk pengkategorian dapat dilihat pada nama serangan eksploitasi.




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

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* dan *cost* dari hulu ke hilir dengan menggunakan perlindungan dari WAF. Ukuran time pada masing masing langkah pada *attack tree* berupa akumulasi waktu dari step mulai hingga selesai. Ukuran cost pada masing masing langkah pada *attack tree* berupa keterangan urutan langkah pada proses penyerangan. Sehingga skor untuk pengkategorian dapat dilihat pada nama serangan eksploitasi. Bentuk perlindungan dari WAF adalah dengan memutuskan koneksi dari penyerang dan memberikan pesan kode atau halaman dengan memuat "403 Forbidden".

## 4. Kesimpulan

Berdasarkan analisa pada bagian sebelumnya, peneitian ini menghasilkan kesimpulan bahwa *Activity diagram* dan *data flow diagram* disusun berdasarkan hasil tahapan eksploitasi untuk menggambarkan *attack tree*. *Attack tree* dapat disusun berdasarkan karakter menggunakan relasi metrik *time* dan *cost*. Metrik *time* dan *cost* dapat digunakan untuk pengkategorian macam macam *attack tree*. Pengkategorian tertinggi tanpa WAF adalah XSS (*Reflected*) dengan skor 53,69. *SQL Injection* menempati urutan terakhir dengan skor 682,49. Dengan kinerja WAF yang melakukan pemutusan eksploitasi, berpengaruh pada pengkategorian *attack tree*. Dengan skor tertinggi XSS (*Reflected*) 61,11. *Brute Force* menempati urutan terakhir dengan skor 207,22.

## Daftar Pustaka

[1] Adem Tekerek, & Omer Faruk Bay. (2019). Design And Implementation Of An Artificial Intelligence-Based Web Application Firewall Model. *Neural Network World*, *29*(4), 189–206. https://doi.org/10.14311/NNW.2019.29.013.
[2] Andria. (2020). Analisis Celah Keamanan Website Menggunakan Tools WEBPWN3R di Kali Linux. *Juli 2020 Generation Journal*, *4*(2). http://www.starrybyte.com.
[3] Sampurna, M. R., Korespondensi, P., Muhammad, :, & Sampurna, R. (2022). Implementasi Hydra, FFUF Dan WFUZZ Dalam Brute Force DVWA. *NetPLG Journal of Network and Computer Applications*, *1*(2).
[4] Namit Gupta, & Abakash Saikia. (2007). *Web Application Firewall*.
[5] Darojat, E. Z., Sediyono, E., & Sembiring, I. (2022). Vulnerability Assessment Website E-Government dengan NIST SP 800-115 dan OWASP Menggunakan Web Vulnerability Scanner. *Jurnal Sistem Informasi Bisnis*, *12*(1), 36–44. https://doi.org/10.21456/vol12iss1pp36-44.
[6] Kuipers, L. (2020). *Analysis of Attack Trees: fast algorithms for subclasses*.
[7] Ingoldsby, T. R. (2009). *Attack Tree-based Threat Risk Analysis*. www.amenaza.com