JIKO (Jurnal Informatika dan Komputer) Akreditasi KEMENRISTEKDIKTI, No.105/E/KPT/2022
Vol. 5, No. 3, Desember 2022, hlm. 224-229 p-ISSN: 2614-8897
DOI: 10.33387/jiko e-ISSN: 2656-1948

DETEKSI DINI SERANGAN PADA *WEBSITE* MENGGUNAKAN METODE ANOMALI BASED

Fariadi[1], M Reza Redo Islami[2]

[1,2]Teknik Informatika, STMIK Dharma Wacana Metro
email: [1]fariadi223@gmail.com, [2]rzredo@gmail.com

(Naskah masuk: 11 November 2022, diterima untuk diterbitkan: 17 November 2022)

Abstrak

Website merupakan sebuah sarana dalam mengakses informasi kapanpun dan dimanapun selain itu *website* juga merupakan media yang digunakan untuk memasarkan produk, mempresentasikan citra perusahaan maupun instansi sehingga perlu di jaga keamanannya, tingginya minat pada *website* dapat meningkatkan resiko *website* dari serangan pihak yang tidak bertanggung jawab *(bad actors)*. Serangan pada *website* dapat mengganggu intergritas data *(integrity)* dan ketersediaan data *(availability)*, untuk itu penelitian ini akan membangun sistem yang dapat mendeteksi dan melakukan antisipasi serangan dengan meggunakan metode anomali based yang akan mendeteksi serangan berdasarkan perilaku *visitor website* dan mengantisipasi serangan tersebut secara otomatis dengan memblokir *IP Address* pengguna, ketika mendeteksi serangan sistem akan menghitung beberapa variable yaitu jumlah *error* 404 *(not found)* yang didapatkan dari catatan pengunjung (*log visitor)*, pengunjung dianggap sebuah anomali apabila pengunjung melakukan hal yang tidak wajar yaitu melebihi batas *error* 404 yang di tentukan dengan melakukan kalkulasi rata-rata *error* 404 dalam satu hari, apabila terdapat pengguna yang melakukan *error* 404 melebihi rata-rata maka *IP Address* pengguna akan di blokir. Berdasarkan pengujian yang dilakukan dengan menggunakan metode *penetration testing*, sistem deteksi menggunakan metode anomali based dapat mendeteksi serangan dengan baik.

Kata kunci: Keamanan siber, Catatan Pengunjung, deteksi intrusi, berbasis-anomali

EARLY DETECTION OF ATTACKS ON WEBSITESUSING THE BASED ANOMALY METHOD

Abstract

The website is a means of accessing information anytime and anywhere besides that the website is also a medium used to market products, and present corporate and agency images so it needs to be kept safe, high interest in websites can increase the risk of websites from attacks by irresponsible parties (bad actors). Attacks on websites can disrupt data integrity and data availability, for this reason, this research will build a system that can detect and anticipate attacks using an anomaly-based method that will detect attacks based on the behavior of website visitors and anticipate these attacks automatically. by blocking the user's IP address, when it detects an attack the system will calculate several variables, namely the number of 404 errors (not found) obtained from visitor logs, visitors are considered an anomaly if visitors do things that are not reasonable, namely exceeding the 404 error limit specified determine by calculating the average 404 error in one day if there are users who make 404 errors more than the average, the user's IP address will be blocked. Based on the tests carried out using the penetration testing method, the detection system using the anomaly-based method can detect attacks well.

Keywords: *Cyber security, Visitor log, intrusion detection, anomaly-based*

1. PENDAHULUAN

Aplikasi berbasis web memiliki ancaman seperti *SQL injection*, aplikasi yang tidak berjalan normal (malfungsi), yang dapat mempengaruhi keseluruhan TI sistem, jika serangan tidak dicegah maka akan mempengaruhi sistem itu sendiri [1]. Aplikasi berbasis *WEB* sendiri memiliki banyak keuntungan dalam penggunaanya yang tidak dimiliki oleh aplikasi konvensional terutama dapat diakses darimanapun dan kapanpun, namun sangat rentan akan kebocoran data [2]. Oleh karena itu serangan pada *website* harus dicegah sedini mungkin sehingga

dapat meningkatan keamanan sistem dan mencegah kebocoran data.

Dalam mengamankan sebuah website ada beberapa cara yang dilakukan, salah satunya adalah dengan memasang *intrusion detection system (IDS)* yang dapat mendeteksi aktifitas yang tidak normal dan memberikan notifikasi sehingga *administrator* dapat memberikan solusi yang terbaik dengan cepat dan efektif [3]. *IDS* sendiri terbagi menjadi dua tipe yaitu *signature-based* dan *anomaly-based*, *anomaly-based* mengklasifikasikan serangan dengan membandingkan paket, apabila paket dirasa tidak normal maka paket akan diklasifikasi sebagai serangan. berbeda dengan *anomaly-based, signature-based* membandingkan paket data dengan data serangan yang telah diidentifikasi sebelumnya, sehingga *anomaly-based* memiliki kelebihan dapat menganalisis serangan yang tidak dikenali sebelumnya [4]. Deteksi anomali dapat dimanfaatkan untuk mengenali serangan dari pengalaman untuk mendeteksi perilaku yang tidak normal dari pengguna [5].

Pengembangan deteksi dan antisipasi seperti pada penelitian [6] telah berhasil mengembangkan *network intrusion prevention system (NIPS*) dengan memanfaatkan *software* Suricata yang menganalisis serangan berdasarkan *packet flow* pada jaringan. Begitupun dengan pemanfaatan *tools* lain yaitu [7], memanfaatkan Wazuh yang mengumpulkan informasi menggunakan *log* aktivitas yang dilakukan oleh agent berupa kegiatan yang dilakukan seperti *create, read, update* dan *delete*. Apabila aktivitas dianggap mencurigakan, wazuh akan memberikan informasi kepada *administrator* melalui *email*.

Anomaly-based dapat digunakan dalam mendeteksi perubahan yang tidak normal pada *website* [8], berhasil mendeteksi anomali yang terjadi akibat perubahan halaman *website* ketika terjadi serangan *defacement*, dengan membandingkan catatan *hash* awal *website* dengan hash terbaru, apabila *hash* tidak singkron mekanisme akan mengirimkan pesan melalui Telegram kepada *administrator website*.

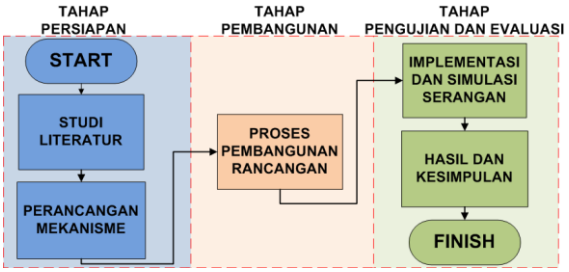
Berdasarkan penelitian [6],[7] dan [8], yang memanfaatkan catatan atau *log* untuk dianalisis sebagai bahan mendeteksi serangan, penelitian ini akan membangun mekanisme deteksi serangan pada website dengan memanfaatkan *visitor log* dibangun menggunakan bahasa pemrograman *hypertext processor* (PHP) dan di integrasikan dengan *database* MySQL, sehingga dapat mendeteksi dan mengantisipasi serangan. Mekanisme yang dibangun tidak melibatkan *administrator* didalam mengantisipasi serangan (otomatis), mengingat serangan dapat terjadi kapan saja.

Deteksi dan antisipasi serangan menggunakan *visitor log* ini berbeda dengan penelitian sebelumnya yang hanya memanfaatkan *visitor log* untuk mendeteksi serangan sedangkan penelitian ini mendeteksi serangan dengan *visitor log* dan melakukan antisipasi dengan melakukan *blocking* terhadap *IP Address* pengguna yang di anggap sebagai sebuah serangan dengan memanfaatkan *file htaccess*.

2. METODE PENELITIAN

Penelitian ini akan dirancang sebuah mekanisme dalam mendeteksi serangan yang terjadi pada sebuah *website* (Apache *webserver*) dengan menggunakan metode *anomaly-based* yang memanfaatkan *visitor log* sebagai bahan analisa dalam menentukan sebuah serangan. Mekanisme yang dibagun akan memanfaatkan bahasa pemrograman *hypertext processor* (PHP) dan menggunakan *Database* MySQL yang sangat umum digunakan didalam sebuah *website* [9].

Adapun langkah yang akan dilakukan dalam penelitian ini dapat ditunjukan pada gambar 1 di bawah ini.



Gambar 1. Alur Penelitian

Keterangan Alur Penelitian:

a. Studi Literatur

Pada studi literatur peneliti akan mempelajari tentang serangan yang terjadi pada sebuah website, pengertian dan penjelasan tentang *anomaly-based* dari penelitian sebelumnya.

b. Perancangan Mekanisme

Pada tahap ini penulis akan merangcang mekanisme dalam mendeteksi serangan dengan memanfaatkan *visitor log*.

c. Proses Pembangunan Rancangan

Setelah rancangan dan mekanisme telah dirancang, maka penulis akan mulai membuat rancangan tersebut menjadi sebuah mekanisme yang siap digunakan.

d. Implementasi dan simulasi serangan

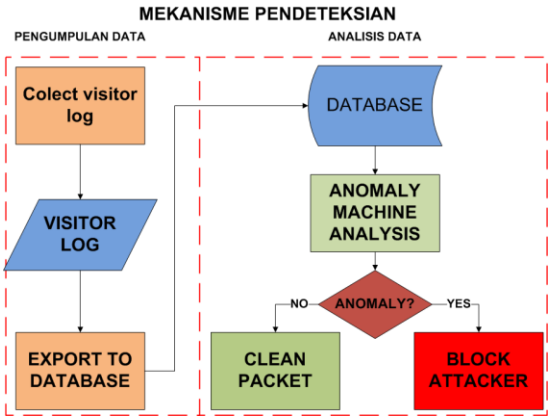
Setelah mekanisme siap digunakan, akan dilakukan *Testing* serangan dengan menggunakan beberapa *tools* untuk mengetahui apakah mekanisme yang dibuat telah berhasil mendeteksi serangan.

e. Hasil dan Kesimpulan

Setelah proses implementasi dan simulasi serangan dilakukan maka akan ditemukan hasil dari serangan tersebut sehingga dapat menyimpulkan apakah mekanisme tersebut telah berhasil dalam mendeteksi dan mengantisipasi serangan.

3. HASIL DAN PEMBAHASAN

Berikut ini adalah alur dari mekanisme pendeteksian dan antisipasi serangan pada *website* yang di tunjukan pada gambar 2 di bawah ini.



Gambar 2. Mekanisme deteksi dan antisipasi

Penjelasan gambar 2 mengenai mekanisme deteksi dan antisipasi serangan akan dijelaskan sebagai berikut.

3.1 Mengumpulkan *visitor log*

Visitor log adalah sebuah catatan yang dihasilkan dari aktivitas pengunjung *website*, *visitor log* dapat dianalisis sebagai bahan pertimbangan dalam menentukan perilaku dari pengunjung *website* [10]. Perilaku pengunjung sangat bermanfaat untuk dianalisa, pengunjung yang mengunjungi *website* secara normal dan tidak normal akan dapat dianalisis melalui perilakunya, *visitor log* berisi waktu kunjungan, *IP address* dari pengunjung, *URL* apa yang di kunjungi dan *response* apa yang diberikan oleh *webserver* kepada pengunjung. pengunjung normal akan membuka *URL* yang tertera pada *website* sehingga akan sangat jarang menemui *error not found* (*error* 404), sedangkan yang tidak normal akan lebih sering melakukan uji coba terhadap *URL* dengan menebak URL dengan value tertentu guna menemukan suatu celah yang ada pada website sehingga akan sering menemui *error* 404 karena *URL* yang diminta tidak ditemukan pada *website*.

[11] menjelaskan bahwa *URL* pada *website* dapat menjadi sebuah titik masuk *(entry point)* untuk dapat di eksploitasi apabila *entry point* tersebut di indikasi sebagai sebuah celah pada *website*. Dari hal tersebut dalam mengumpulkan *visitor log* terdapat beberapa variabel yang akan digunakan sebagai bahan analisa untuk deteksi sebuah *anomaly* pada *website*.

Berikut ini tabel 1 merupakan beberapa variabel yang akan di catat pada *visitor log*.

Tabel 1: Detail *Visitor log*

Kolom	**Detail**
time	Berisikan waktu *visitor* mengakses *URL*
source	*IP address visitor*
URL	Link yang diakses oleh *visitor*
response	Response server terhadap request *visitor*

Dalam mengumpulkan *visitor log* diperlukan *script hypertext processor* (PHP), *script* ini difungiskan untuk menangkap beberapa variabel yang dibutuhkan untuk dianalisis, untuk mengumpulkan variabel tersebut *file* ini harus di panggil ke setiap halaman yang ada di *website*, berikut ini adalah gambar 3 yang ber isi *script* dari *file* pencatat *visitor log*.

```php
<?php
#variabel file
$file = "log.txt";

#pembuatan file
$f=fopen($file,'a');
$time = date('d-m-Y H:i:s', $_SERVER['REQUEST_TIME']); # waktu request
$source = $_SERVER['REMOTE_ADDR']; # IP address pengunjung
$reqURL = $_SERVER['REQUEST_URI']; # URL yan dikunjungi
$response = http_response_code(); # response server atas request pengunjung

fwrite($f, $time.";". $source.";".$reqURL.";".$response. "\n");
fclose($f);
?>
```

Gambar 3. Generate *file log*

Script pada gambar 3 akan dijalankan apabila pengunjung membuka sebuah halaman yang ada pada *website*, sehingga akan membuat *file log*.txt yang tersimpan di *website*.

Berikut ini adalah gambar 4 yaitu *file log*.txt yang dihasilkan *script generate file log*.

```
05-10-2022 14:55:22;172.70.188.51;/anomaly/;200
09-10-2022 13:21:13;172.70.147.198;/anomaly/;200
09-10-2022 13:21:28;162.158.163.176;/anomaly/;200
```

Gambar 4. *Visitor log* dengan format txt

3.2 Import data *visitor* ke database

Log yang di hasilkan dari proses sebelumnya akan di *import* ke dalam *database*, Proses *import data visitor log* dilakukan dengan cara menjalankan *file* importer.php secara otomatis dengan memanfaatkan *cronjob* yang dijalankan dengan interval setiap 5 menit sekali.

file importer.php ini bertugas untuk membaca *log*.txt dan membentuknya menjadi variabel yang akan di *import* ke dalam tabel *flow*. Sehingga dapat dilakukan seleksi dengan menjalankan query database MySQL menggunakan bahasa pemrograman *PHP*.

berikut ini gambar 5 yang merupakan isi dari *file* importer.php.

```php
<?php
include"config/koneksi.php"; #koneksi ke database

$file = fopen("log.txt","r"); #membuka file log.txt

#memecah file menjadi isi variabel
while(!feof($file)) {
    $getTextline = fgets($file);
    $explodeline = explode(";",$getTextline);
    list($time,$source,$url,$response) = $explodeline;

# insert ke database
$query= mysql_query("insert into flow (time,source,url,response)
values ('$time', '$source', '$url', '$response')") or die (mysql_error());
}
fclose($file);

?>
```

Gambar 5. *File* importer *log* ke database

Berikut ini gambar 6 yang ber isi struktur dari tabel *flow* yang ada di *database* anomali.

	#	Name	Type	Collation
☐	1	id_flow	int(11)	
☐	2	time	varchar(255)	latin1_swedish_ci
☐	3	source	varchar(255)	latin1_swedish_ci
☐	4	url	text	latin1_swedish_ci
☐	5	response	text	latin1_swedish_ci

Gambar 6. Struktur tabel flow

Setelah *script* importer.php di jalankan, maka table flow akan terisi sesuai yang ada pada *log*.txt. berikut ini adalah gambar 7 yang menunjukan berhasilnya proses import *log*.txt ke dalam tabel flow.

+ Options

	id_flow	time	source	url	response
Edit Copy Delete	118	05-10-2022 14:55:22	172.70.188.51	/anomaly/	200
Edit Copy Delete	119	09-10-2022 13:21:13	172.70.147.198	/anomaly/	200
Edit Copy Delete	120	09-10-2022 13:21:28	162.158.163.176	/anomaly/	200

Gambar 7. *Visitor log* berhasil di import ke database

3.3 Deteksi dan antisipasi serangan

Data yang telah di *import* ke dalam *table flow* akan di analisis dengan menggunakan *script PHP* yang akan melakukan penentuan apakah seorang *visitor* melakukan serangan atau tidak.

Berikut ini gambar 8 yang menggambarkan isi *script* detection.php.

```php
#MEMILIH WAKTU SEKARANG
$sekarang = mysql_query("SELECT CURRENT_DATE() as sekarang");
$date = mysql_fetch_array($sekarang);
$hari = $date['now']."00:00:00";
echo"#DEFAULT HTACCESS \n";
echo"ErrorDocument 404 https://gwex.tech/anomaly/notfound.php \n";

#QUERY SELECTION URL BRUTEFORCE
$sourceURL = mysql_query("SELECT * FROM flow WHERE url='/anomaly/notfound.php'
AND time >='$hari' having count(*) > 20 ");
echo"#BRUTEFORCE URL ATTACK \n";
while ($ip = mysql_fetch_array($sourceURL)) {
   echo"Deny from $ip[source] \n ";
};

#QUERY SELECTION LOGIN BRUTEFORCE
$sourceLogin = mysql_query("SELECT * FROM flow WHERE url='/anomaly/loginerror.php'
AND time >='$hari' having count(*) > 5 ");
echo"#BRUTEFORCE LOGIN ATTACK \n";
while ($ip = mysql_fetch_array($sourceLogin)) {
   echo"Deny from $ip[source] \n ";
};
```

Gambar 8. *File* yang digunakan untuk membentuk htaccess

Script detection.php akan di jalankan menggunakan *cronjob*, yang akan membentuk *file* .htaccess. berikut ini adalah tabel 2 ber isi parameter yang digunakan dalam mendeteksi dan mengelompokan serangan.

Tabel 2: Parameter serangan

SERANGAN	INFO	TRESHOLD
URL Bruteforce	Serangan ini dilakukan dengan menebak *URL* baik untuk mencari bug SQL injection maupun untuk mencari halaman *log*in	20
LOGIN Bruteforce	Serangan ini dilakukan dengan menebak *username* dan *username*.	5

Script akan melakukan seleksi terhadap *IP address (source)* yang telah melakukan kesalahan lebih dari *threshold* yang telah ditentukan, apabila terdapat *IP address* yang melakukan kesalahan melebihi *threshold* maka *IP address* tersebut akan di blokir dengan melakukan *update file* htaccess. yaitu *file* yang digunakan untuk memproteksi *file* atau *directory* pada *website* [12].

Treshold yang di dapat merupakan gambaran pengunjung normal, sehingga apabila terdapat pengguna yang melakukan *error 404* melebihi threshold dianggap sebagai *point anomalies* [13] yaitu penyimpangan individual dari *behavior* pengunjung.

3.4 *Testing* serangan

Testing serangan akan dilakukan secara manual (tanpa menggunakan *tool*), serangan ini akan terbagi menjadi dua bagian yaitu serangan *URL bruteforce* yang digunakan untuk menemukan halaman admin sedangkan serangan berikutnya akan dilakukan *bruteforce log*in *administrator*.

Proses *testing* ini juga dapat disebut sebagai penetrasi *testing* [14] yaitu metode evaluasi keamanan sistem komputer atau jaringan komputer dengan melakukan simulasi serangan.

a. *Testing* Serangan *Log*in *Bruteforce*

Bruteforce attack merupakan metode menebak sebuah *password* dengan menggunakan kata yang sering di gunakan hingga menemukan *password* pada sistem tertentu [15]. *Testing* dilakukan dengan memasukan *username* dan *password* admin secara *random*, apabila *username* dan *password* tidak sesuai maka *website* akan mengarah ke halaman loginerror.php, sehingga aktifitas akan dicatat oleh mekanisme yang telah di bangun dan akan di simpulkan sebagai serangan apabila kegagalan *login* melebihi ambang batas yang telah ditentukan (*threshold*).

b. *Testing* Serangan *URL Bruteforce*

Testing serangan akan dilakukan dengan mencoba mencari *URL* halaman admin dengan memasukan *URL* ke dalam *browser* yang identik dengan halaman admin, serangan akan berhasil apabila halaman admin ditemukan. Namun apabila *URL* tidak ditemukan maka *user* akan diarahkan ke halaman notfound.php sehingga aktifitas tersebut akan dicatat oleh mekanisme yang telah dibangun dan akan di simpulkan sebagai serangan apabila telah melebihi ambang batas yang telah ditentukan (*threshold*).

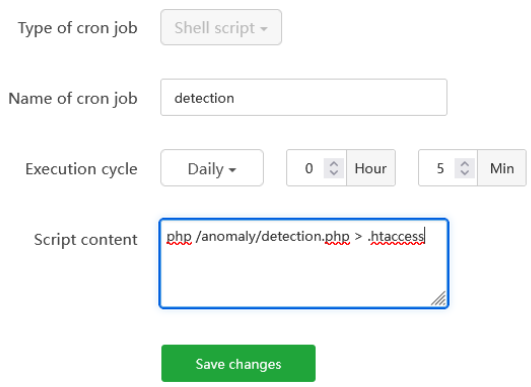
3.4 Hasil *Testing* serangan

Dari hasil uji coba yang telah dilakukan mekanisme telah berhasil mencatat aktifitas *visitor* seperti yang ditunjukan oleh gambar 9.

Gambar 9. *log* serangan di import ke table flow

Visitor log akan di *import* ke *table flow* pada *database* anomali oleh *script* importer.php yang berjalan otomatis dengan fitur *cronjob*, sehingga detection.php akan melakukan seleksi *IP address* (*source*) yang melebihi *threshold*, dan melakukan *editing* pada *file* .htaccess untuk memblokir *IP address* yang melanggar *rule*.

Berikut ini gambar 10 berisi *cronjob* yang diajalankan untuk melakukan *editing* pada *file* htaccess.



Gambar 10. *Cronjob create* .htaccess

Setelah *cronjob* di jalankan maka akan terbentuk *file* htaccess yang berisikan informasi untuk melakukan blokir terhadap *IP address* yang telah diseleksi sebagai serangan.

Berikut ini gambar 11 yang berisi *file* htaccess yang terbentuk dari proses seleksi.

Gambar 11. isi *file* .htaccess

File htaccess yang telah dibuat akan memblokir *IP address (source)* yang teridentifikasi sebagai sebuah anomali atau suatu kegiatan yang dianggap tidak normal dari pengunjung, yaitu *IP address* yang melakukan kesalahan dalam mengakses *URL* dan kesalahan dalam *login* melebihi *threshold* yang telah ditentukan.

4. KESIMPULAN

Mekanisme deteksi dan antisipasi serangan telah berjalan dengan baik dan dapat mendeteksi serangan berupa *bruteforce URL* dan *bruteforce* login admin. Kecepatan deteksi ditentukan oleh interval yang dijalankan oleh *cronjob*. Pendeteksian dengan *visitor log* memiliki kelebihan dimana dapat melindungi website yang menggunakan *protocol HTTPS* maupun *HTTP*. Mekanisme yang dibangun tidak membutuhkan campur tangan *administrator* dalam mengantisipasi serangan. Sehingga dapat melindungi *website* secara *realtime*.

5. DAFTAR PUSTAKA

[1] Lella, I. et al. (2021) “ENISA Threat Landscape 2021.”

[2] Rafeli, A. I., Seta, H. B. and Widi, I. W. (2022) “Pengujian Celah Keamanan Menggunakan Metode OWASP Web Security Testing Guide (WSTG) pada Website XYZ,” *Informatik : Jurnal Ilmu Komputer*, 18(2), p. 97. doi: 10.52958/iftk.v18i2.4632.

[3] Díaz-Verdejo, J. et al. (2022) “On the detection capabilities of signature-based Intrusion Detection Systems in the context of web attacks,” *Applied sciences (Basel, Switzerland)*, 12(2), p. 852. doi: 10.3390/app12020852.

[4] Lin, I.-C., Chang, C.-C. and Peng, C.-H. (2022) “An anomaly-based IDS framework using centroid-based classification,” *Symmetry*, 14(1), p. 105. doi: 10.3390/sym14010105.

[5] Martins, I. et al. (2022) “Host-based IDS: A review and open issues of an anomaly detection system in IoT,” *Future generations computer systems: FGCS*, 133, pp. 95–113. doi: 10.1016/j.future.2022.03.001.

[6] Tanang Anugrah, F., Ikhwan, S. and Gusti A.G, J. (2022) “Implementasi Intrusion Prevention System (IPS) Menggunakan Suricata Untuk Serangan SQL Injection,” *Techné Jurnal Ilmiah Elektroteknika*, 21(2), pp. 199–210. doi: 10.31358/techne.v21i2.320.

[7] Nova, F., Pratama, M. D. and Prayama, D. (2022) "Wazuh sebagai Log Event Management dan Deteksi Celah Keamanan pada Server dari Serangan Dos," *JITSI : Jurnal Ilmiah Teknologi Sistem Informasi*, 3(1), pp. 1–7. doi: 10.30630/jitsi.3.1.59.

[8] Indah Septiani, N. et al. (2022) "PERANCANGAN WEB DEFACEMENT MONITORING DENGAN MENGGUNAKAN METODE KOMPARASI NILAI HASH1," *JIKO (Jurnal Informatika dan Komputer)*, 5(2), pp. 150–155. doi: 10.33387/jiko.v5i2.4852.

[9] Saputra, I. P., Yusuf, R. and Saprudin, U. (2021) "IMPLEMENTASI CLOUD COMPUTING SEBAGAI RADIUS SERVER PADA JARINGAN INTERNET ROUTER MIKROTIK," *Journal Computer Science and Informatic Systems : J-Cosys*, 1(2). doi: 10.53514/jc.v1i2.67.

[10] Lopes, P. and Roy, B. (2015) "Dynamic recommendation system using web usage mining for E-commerce users," *Procedia computer science*, 45, pp. 60–69. doi: 10.1016/j.procs.2015.03.086.

[11] Huang, H. C. et al. (2022) "Adaptive Entry Point Discovery for Web Vulnerability Scanning," *Journal of Information Science & Engineering*, (1).

[12] Verma, L. (2021) "OJS security analysis issues, reasons, and possible solutions," *DESIDOC journal of library and information technology*, 41(5), pp. 391–396. doi: 10.14429/djlit.41.5.15975.

[13] Imam, R. M., Sukarno, P. and Nugroho, M. A. (2019) "Deteksi Anomali Jaringan Menggunakan Hybrid Algorithm," *eProceedings of Engineering*, 6(2).

[14] Sahren, S., Dalimuthe, R. A. and Amin, M. (2019) "Penetration Testing Untuk Deteksi Vulnerability Sistem Informasi Kampus," *Prosiding Seminar Nasional Riset Information Science (SENARIS)*, 1, p. 994. doi: 10.30645/senaris.v1i0.109.

[15] Gautam, T. and Singh, U. (2022) AN APPROACH FOR DETECTING PASSWORD PATTERN IN DICTIONARY ATTACK.