Fariadi

anonymous marking enabled

Submission date: 13-Nov-2022 08:45PM (UTC-0500)
Submission ID: 1940234001
File name: 5352-14051-1-RV.docx (357.95K)
Word count: 2078
Character count: 13480

JIKO (Jurnal Informatika dan Komputer) Akreditasi KEMENRISTEKDIKTI, No.36/E/KPT/2019
Vol. x, No. x, April 2019, hlm. x-x 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]Teknik Informatika, STMIK Dharma Wacana Metro
[2] Teknik Informatika, STMIK Dharma Wacana Metro
Email: [1]fariadi223@gmail.com, [2]rzredo@gmail.com

(Naskah masuk: ddmmm yyyy, diterima untuk diterbitkan: ddmmmyyyy)
(1 bariskosong, 10pt)

Abstrak

Website merupakan sebuah sarana dalam mengakses informasi kapanpun dan dimanapun, tingginya minat pada website dapat meningkatkan resiko website untuk diserang. Serangan pada website dapat mengganggu data integrity dan availability, untuk itu penelitian ini akan membangun sistem pendeteksian serangan dengan meggunakan metode anomali based yang akan mendeteksi serangan berdasarkan perilaku visitor website, dalam mendeteksi serangan sistem akan menghitung beberapa variable yang didapatkan dari log visitor, visitor dianggap anomali jika visitor melakukan hal yang tidak wajar. Berdasarkan pengujian yang dilakukan dengan menggunakan metode penetration Testing, sistem deteksi menggunakan metode anomali based dapat mendeteksi serangan dengan baik.

Kata kunci: *security, Visitor log, intrusion detection, anomali-based*

EARLY DETECTION OF ATTACKS ON WEBSITESUSING THE BASED ANOMALY METHOD

Abstract

Websites are a means of accessing information and hopefully, interest in websites can increase the risk of websites being attacked. Attacks on websites can disrupt the integrity and availability of data, this research will build an attack detection system using an anomali-based method that will detect attacks based on the behavior of website visitors, in detection the system will calculate several variables obtained from visitor logs, visitors are considered an anomali if a visitor does something unnatural. Based on the tests conducted using the penetration Testing method, the detection system using the anomali-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 anomali-based, anomali-

based mengklasifikasikan serangan dengan membandingkan paket, apabila paket dirasa tidak normal maka paket akan diklasifikasi sebagai serangan. berbeda dengan anomali-based, signature-based membandingkan paket data dengan data serangan yang telah diidentifikasi sebelumnya, sehingga anomali-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].

Penelitian [6], berhasil mengembangkan network intrusion prevention system (NIPS) dengan memanfaatkan software Suricata yang menganalisis serangan berdasarkan packet flow pada jaringan.

Penelitian [7], memanfaatkan Wazuh yang mengumpulkan informasi menggunakan log aktivitas yang dilakukan oleh agent berupa kegiatan yang dilakukan seperti create, read, updte dan delete. Apabila aktivitas dianggap mencurigakan, wazuh akan memberikan informasi kepada *administrator* melalui email.

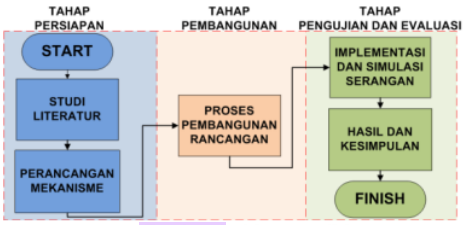
Penelitian [8], berhasil mendeteksi anomali yang terjadi akibat perubahan halaman website ketika terjadi serangan defacement, dengan membandingkan catatan hash awal website dengan hash terbaru, apabila hash tidak singkron mekanisme akan mengirimkan pesan melalui Telegram kepada *administrator* website.

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 diintegrasikan 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.

2. METODE PENELITIAN

Pada penelitian ini penulis akan merancang sebuah mekanisme dalam mendeteksi serangan yang terjadi pada sebuah website (Apache webserver) dengan menggunakan metode anomali-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 adalah sebagai berikut.



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 anomali-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

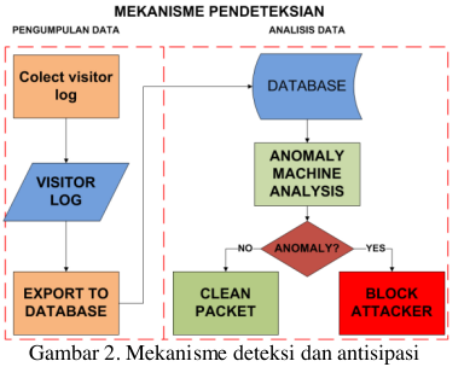
setelah mekanisme siap digunakan, akan dilakukan Testing serangan dengan menggunakan beberapa tool untuk mengetahui apakah mekanisme yang dibuat telah berhasil mendeteksi serangan.

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. PERANCANGAN SYSTEM

Berikut ini adalah alur dari mekanisme pendeteksian dan antisipasi serangan pada *website*.



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, pengunjung normal akan membuka *URL* yang tertera pada *website* sehingga jarang menemui *error not found* (*error 404*), sedangkan yang tidak normal akan lebih sering melakukan uji coba terhadap *URL* dan akan sering menemui *error* 404 karena *URL* yang diminta tidak ditemukan pada *website*. Dalam penelitian ini penulis mengambil beberapa variabel yang ada pada *visitor log* untuk dianlisis, berikut ini adalah variabel yang akan dianalisis.

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 isi 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 di *website*, sehingga akan membuat *file log*.txt yang tersimpan di *website*. Berikut ini adalah isi dari *log*.txt yang dihasilkan *script* gambar 3.

```
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

Proses import data *visitor log*, adalah dengan menjalankan *file* importer.php dengan memanfaatkan cronjob yang dijalankan dengan interval setiap 5 menit sekali, *file* ini bertugas untuk membaca *log*.txt dan membentuknya menjadi variabel yang akan diimport ke dalam tabel flow, berikut ini adalah 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 adalah 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 dijalankan, maka table flow akan terisi sesuai yang ada pada *log*.txt. berikut ini adalah gambar 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 dianalisis dengan menggunakan *script* PHP yang akan melakukan penentuan apakah seorang *visitor* melakukan serangan atau tidak.
Berikut ini isi *script* detection.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 dijalankan menggunakan *cronjob*, yang akan membentuk *file* .htaccess. berikut ini adalah tabel 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
*LOG*IN *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 mengupdate *file* htaccess. yaitu *file* yang digunakan untuk memproteksi *file* atau *directory* pada *website* [11].

3.4 *Testing* serangan

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

a. *Testing* Serangan *Login Bruteforce*

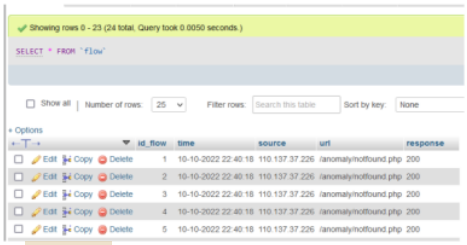
Testing dilakukan dengan memasukan *username* dan *username* admin secara random, apabila *username* dan *username* tidak sesuai maka *website* akan mengarah ke halaman loginerror.php sehingga aktifitas akan dicatat oleh mekanisme yang telah dibangun dan akan disimpulkan sebagai serangan apabila kegagalan *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 disimpulkan sebagai serangan apabila telah melebihi ambang batas yang telah ditentukan (*threshold*).

4. HASIL DAN PEMBAHASAN

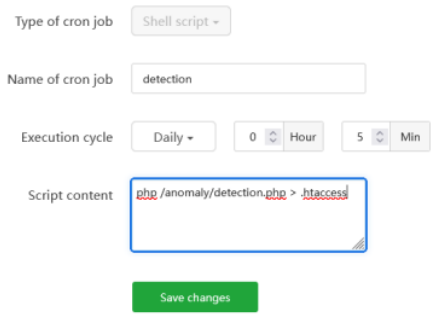
Dari hasil uji coba yang telah dilakukan mekanisme telah berhasil mencatat aktifitas *visitor* seperti yang ditunjukan oleh gambar 9.



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

Visitor log akan diimport 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 *cronjob* yang diajalankan untuk melakukan *editing* pada *file* htaccess.



Gambar 10. *Cronjob create* .htaccess

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

5. 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*.

6. DAFTAR PUSTAKA

[1] I. Lella, Marianthi Theocharidou, E. Tsekmezoglou, and Apostolos Malatras, *ENISA Threat Landscape 2021*. 2021.

[2] A. I. Rafeli, H. B. Seta, and I. W. Widi, "Pengujian Celah Keamanan Menggunakan Metode OWASP Web Security Testing Guide (WSTG) pada *Website* XYZ," *Informatik : Jurnal Ilmu Komputer*, vol. 18, no. 2, p. 97, Aug. 2022, doi: 10.52958/iftk.v18i2.4632.

[3] J. Díaz-Verdejo, J. Muñoz-Calle, A. Estepa Alonso, R. Estepa Alonso, and G. Madinabeitia, "On the Detection Capabilities of Signature-Based Intrusion Detection Systems in the Context of Web Attacks," *Applied Sciences*, vol. 12, no. 2, p. 852, Jan. 2022, doi: 10.3390/app12020852.

[4] I.-C. Lin, C.-C. Chang, and C.-H. Peng, "An Anomali-Based IDS Framework Using Centroid-Based Classification," *Symmetry*, vol. 14, no. 1, p. 105, Jan. 2022, doi: 10.3390/sym14010105.

[5] I. Martins, J. S. Resende, P. R. Sousa, S. Silva, L. Antunes, and J. Gama, "Host-based IDS: A review and open issues of an anomali detection system in IoT," *Future Generation Computer Systems*, vol. 133, pp. 95–113, Aug. 2022, doi: 10.1016/j.future.2022.03.001.

[6] Tanang Anugrah, F., Ikhwan, S., & Gusti A.G, J. (2022, September 29). Implementasi Intrusion Prevention System (IPS) Menggunakan Suricata Untuk Serangan SQL Injection. *Techné : Jurnal Ilmiah Elektroteknika*, *21*(2), 199–210. https://doi.org/10.31358/techne.v21i2.320.

[7] Fitri Nova, Pratama, M. D., & 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), 1–7. https://doi.org/10.30630/jitsi.3.1.59.

[8] Indah Septiani, N., Sediyono, A., & Rochman, A. (2022). Perancangan web defacement monitoring Dengan Menggunakan metode Komparasi Nilai Hash1. JIKO (Jurnal Informatika Dan Komputer), 5(2), 150–155. https://doi.org/10.33387/jiko.v5i2.4852.

[9] I. P. Saputra, R. Yusuf, and U. Saprudin, "IMPLEMENTASI CLOUD COMPUTING SEBAGAI RADIUS SERVER PADA JARINGAN INTERNET ROUTER MIKROTIK," *Journal Computer Science and Informatic Systems : J-Cosys*, vol. 1, no. 2, Jul. 2021, doi: 10.53514/jc.v1i2.67.

[10] Lopes, P., & Roy, B. (2015). Dynamic recommendation system using web usage mining for e-commerce users. Procedia Computer Science, 45, 60–69. https://doi.org/10.1016/j.procs.2015.03.086.

[11] Verma, L. (2021). OJS security analysis issues, reasons, and possible solutions. DESIDOC Journal of Library & Information Technology, 41(5), 391–396. https://doi.org/10.14429/djlit.41.5.15975.

Fariadi

ORIGINALITY REPORT

6%	5%	0%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.coursehero.com Internet Source	2%
2	Submitted to Universitas Islam Lamongan Student Paper	1%
3	www.slideshare.net Internet Source	1%
4	jurnal.stikom.edu Internet Source	<1%
5	eprints.mdp.ac.id Internet Source	<1%
6	eprints.uny.ac.id Internet Source	<1%
7	repository.iainpurwokerto.ac.id Internet Source	<1%
8	scholarhub.ui.ac.id Internet Source	<1%
9	stmik-dw-m.blogspot.com Internet Source	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		

Fariadi