REAL TIME ACCESS LOG ANALYZER FOR DETECTION AND PREVENTION OF WEB ATTACK

Poby Zaarifwandono.[1], Eko Sakti P[2],Ari Kusyanti[3]
Fakultas Ilmu Komputer
Universitas Brawijaya
Jl. Veteran No.8, Malang 65145, Jawa Timur, Indonesia
Pobyzaarif34@gmail.com[1],ekosakti@ub.ac.id[2],ari.kusyanti@ub.ac.id[3]

Abstrak

Website atau situs adalah suatu halaman yang saling terhubung berisi sekumpulan informasi yang disediakan perorangan, kelompok atau suatu organisasi. Setiap *website* juga menyediakan informasi yang berbeda sesuai tipe *website*, seperti *blog* atau personal, bisnis, pemerintahan dan lain sebagainya. Banyaknya kasus pembobolan *website* dan pencurian data menjadi dasar penelitian ini. Penelitian ini berfokus kepada pendeteksian serangan terhadap *website*. *Access log* adalah kumpulan aktivitas *website* berupa alamat *ip*, tanggal, metode *request, request URL*, kode status, jumlah *byte, referrer* dan *user agent.* Dengan menganalisa *access log* beberapa serangan *website* dapat ditemukan tentunya secara otomatis mengingat jumlah baris/*log* satu *file access log* sangat banyak. *Hansipy/log analyzer* dibuat menggunakan bahasa pemrograman python memanfaatkan *library* yang sudah disediakan. *Access log* yang telah didapat akan diolah meliputi pengumpulan data, ekstraksi, analisa dan deteksi. Setelah proses tersebut dilakukan penulisan ulang *log* yang terdekteksi berpotensi serangan kedalam database dan ditampilkan kembali dalam *web report.* Berdasarkan hasil penelitian dapat disimpulkan bahwa sistem *log analyzer/hansipy* dapat diimplementasikan dalam webserver sebagai *Intrusion Detection System/IDS* untuk menjaga dari tindak kriminal di dunia digital khususnya *website.*

Kata kunci: *website, access log, log analyzer*.

Abstract

Website or simply called 'site' is a page that contains a set of interconnected information provided by an individual, a group or an organization. Each website also provides different information according to the types of websites, such as blogs or personal, business, government and others. The number of cases of burglary and theft of data website is the basis of this study. This study focused on the detection of attacks against websites. Access log is a collection of website activity such as IP address, date, request method, request URL, status code, number of bytes, referrer and user agent. By analyzing the access log, some attacks against the website can be found by automatic way, considering the number of rows/log in one log file access can be very much. Hansipy / log analyzer built using python which utilizing libraries that have been provided. Access logs that had been obtained will be processed, including data collection, extraction, analysis and detection. After the process, the detected log which potentially attacked will be rewritten into the database and displayed in the web report. Based on the results of this study concluded that the system log analyzer/hansipy can be implemented in webserver as Intrusion Detection System/IDS to protect from crime in the digital world, especially websites.

Keywords: *website, access log, log analyzer*.

1. Pendahuluan

Dikutip dari website hackmageddon.com yang sudah melakukan survey perbandingan motif *cyber attack* pada tahun 2014 dan 2015. Selama tahun 2015, persentase kejadian termotivasi oleh *Cyber Crime* telah meningkat dari 62,3% menjadi 67%, sedangkan *hacktivism* telah kehilangan tiga poin (20,8% pada tahun 2015 *vs* 24,9 pada tahun 2014). Dan untuk motif *Spionase* Cyber pertumbuhan kecil, sedangkan *Cyber Warfare* dasarnya Stabil. Selain itu data dari www.zone-h.org(*archive defaced websites*) menunjukkan ada sekitar 200 website yang dibobol/hack oleh perorangan atau team dengan motif yang beragam perharinya.

Cyber attack khususnya terhadap website dapat dilakukan dengan beberapa teknik, diantaranya berikut 10 teknik serangan *website* yang dapat dideteksi melalui log pada tahun 2008 berdasarkan jurnal berjudul *"Detecting Attacks on Web Applications from Log". Cross Site Scripting(XSS), Injection Flaws, Malicious File Execution, Insecure Direct Object Reference, Cross Site Request Forgery(CSRF), Information Leakage and Improper Error Handling, Broken Authentication and Session Management, Insecure Cryptographic Storage,*

Insecure Communications, Failure to Restrict URL Access (Meyer, 2008).

Serangan terhadap website telah menciptakan ancaman besar, baik dalam mempertahankan jaringan lokal dan global. Serangan menjadi lebih canggih dengan teknik yang baru dan kecepatan *hacker* melakukan penetrasi terhadap website. Selain itu kelalaian admin dalam me-*maintenace* atau monitoring website menjadi faktor utama dibeberapa kasus *hacking website*.

Berdasarkan permasalahan tersebut penting untuk menyediakan sistem yang diperlukan dalam mendeteksi, mengklasifikasi, dan mempertahankan dari berbagai jenis serangan. Dalam penelitian ini sistem dibuat menggunakan *library python* yang akan melakukan *parsing* terhadap file *access.log* dari *apache server* lalu melakukan analisa pencocokan(*string matching*) ciri/pola serangan terhadap *access log* dan apabila terjadi kecocokan maka sistem akan melakukan pemberitahuan/*report* kepada admin website untuk dengan menuliskan kembali ke dalam database berupa alamat *ip*/*host*, tanggal dan waktu, *method*, serta *request url*.

Diharapkan dari penelitian ini didapat sistem yang mampu mempermudah admin *website* dalam memonitoring *website* dan mengurangi tingkat kriminal di *internet*. Parameter keberhasilan penelitian ini jika sistem dapat mendeteksi adanya serangan dan mengirimkan *log* dan menyimpan ke dalam *database* lalu ditampilkan kembali dalam bentuk *html*. Adapun pemilihan *server*, metode, percobaan serangan apa yang akan dilakukan dalam penelitian ini akan dijelaskan pada bab selanjutnya.

2. Kajian Pustaka

Makalah dari SANS Institute oleh Roger Meyer yang berjudul "*Detecting Attacks on Web Applications from Log Files*". Dalam makalah ini menjelaskan bagaimana cara mendeteksi mendeteksi kelemahan dan keamanan pada aplikasi website. pengolahan log berupa analisis yang lebih rinci dari request pengguna. Disebutkan juga dalam makalah bahwa sebagian besar serangan dapat dikenali dan ditindaklanjuti untuk mencegah eksploitasi lebih lanjut. Dalam makalah ini didapat kesimpulan sebagai berikut:

Ada dua metode deteksi serangan yaitu *rule-based*(aturan statis) dan *anomaly-based*(aturan dinamis). analisis *web server log* menggunakan *rule-based* berkonsentrasi pada serangan web yang terlihat di web bawaan file log server seperti Apache atau IIS.

Ada beberapa hal yang harus diatasi untuk mendeteksi serangan dalam file log. Pertama, pola/*pattern* serangan harus diketahui untuk membuat aturan deteksi. Oleh karena itu, penting untuk mengetahui sebanyak-banyaknya varian serangan yang berbeda. Hal lain yang tak kalah penting adalah varian *encoding* dan standar yang berbeda.

Setelah pola/*pattern* yang berbeda dipelajari, serangan umum(paling sering digunakan) dapat dengan mudah dideteksi. Dapat didefinisikan menggunakan ekspresi regular/*regex* memungkinkan identifikasi dari banyak kelemahan keamanan aplikasi *web* yang paling sering digunakan, dalam hal ini dapat dilihat pada makalah OWASP Top Ten(Meyer, 2008).

Makalah sari SANS Institute oleh Niklas sarokaari yang berjudul "*How to identify malicious HTTP Requests*". Dalam makalah ini menjelaskan tentang pentingnya bagi admin sistem untuk mengetahui bagaimana *request* berbahaya dilakukan dan bagaimana lalu lintas semacam ini dapat diidentifikasi. Ketika aplikasi web dieksploitasi atau sudah dirusak oleh *hacker*, penting untuk menemukan permintaan berbahaya dari log server dan mengidentifikasi apa jenis serangan yang digunakan untuk mengidentifikasi kerentanan dalam aplikasi web. Dalam makalah juga memberikan analisis mendalam tentang penggunaan aplikasi seperti *wireshark appendix*, *beEF* dan *mutillidae* sebagai simulator *web server* untuk di lakukan penetrasi serangan. Dalam makalah ini didapat kesimpulan sebagai berikut :

Hal yang paling umum pada kerentanan keamanan pada aplikasi *website* biasanya terdapat pada validasi input pengguna, penerapan kontrol akses dan mekanisme otentikasi. *Vendor*/admin *website* harus mengerti akan pentingnya hal ini, bahwa seorang hacker dapat mengekploitasi dari kelemahan yang terdapat pada website untuk mengambil informasi sensitif atau mendapat akses yang tidak sah ke aplikasi website. *Bypass* atau bahkan memanfaatkan celah *cross-site scripting*. Menerapkan kontrol keamanan *web* dapat mengurangi resiko terhadap pencurian data dan dapat menjaga kerahasiaan data pengguna.

3. *IDS/Intrusion Detection System*

Intrusion Detection System(IDS) adalah sebuah aplikasi perangkat lunak yang memonitor jaringan atau kegiatan sistem dan mendeteksi adanya gangguan/serangan terhadap sistem utama yang menjalankan layanan (Vijayarani, S.; Sylvia, Maria. 2015). Ibarat rumah adalah sistem yang harus dijaga *IDS* adalah satpam atau sesuatu yang bisa mendeteksi ketika ada maling yang sedang masuk kedalam rumah tersebut. *IDS* sangat penting untuk mencegah terjadinya tindak krimal pada sebuah sistem. *IDS* mempermudah seseorang untuk memonitoring sistem. Seperti gambaran sebelumnya bahwa seseorang juga tidak akan menjaga sebuah rumah selama 24 jam, maka dari itu dibutuhkan sebuah sistem yang otomatis untuk melakukan monitoring.

Intrusion Detection System berdasar kepada apa yang dianalisis diklasifikasikan menjadi tiga kategori yaitu *Host Based IDS*, *Network Based IDS*, *Application Based IDS*. Sedangkan untuk model dasar yang digunakan untuk menganalisis

diklasifikasikan menjadi dua kategori yaitu *Anomaly Based IDS, Signature Based IDS*.

4. Python

Guido van Rossum menciptakan bahasa pemrograman *Python* di akhir 1980-an. Berbeda dengan bahasa populer lainnya seperti *C, C++*, *Java*, dan *C#*, *Python* berusaha untuk menyediakan sintaks yang sederhana namun *powerful*.

Python digunakan untuk pengembangan perangkat lunak di perusahaan dan organisasi seperti *Google*, *Yahoo*, *CERN*, *industrial Light and Magic*, dan *NASA*. programmer berpengalaman dapat mencapai hal-hal besar dengan *Python*, tapi *Python* juga cocok untuk programmer pemula dan memungkinkan mereka untuk mengatasi masalah yang menarik lebih cepat daripada bahasa lainnya, bahasa yang lebih kompleks yang menuntut pemahaman teori dan praktikal(Halterman, 2011).

Sisi utama yang membedakan *Python* dengan bahasa lain adalah dalam hal aturan penulisan kode program. Bagi para programmer di luar python siap-siap dibingungkan dengan aturan inisialisasi, tipe data, pemanggilan fungsi, dan dictionary. Python memiliki kelebihan tersendiri dibandingkan dengan bahasa lain terutama dalam hal penanganan modul, ini yang membuat beberapa programmer menyukai python.

5. Web Attack

Serangan berbasis web dianggap oleh para ahli keamanan menjadi ancaman besar dan paling sedikit dipahami dari semua risiko *cyber attack* yang terkait dengan kerahasiaan (*confidentiality*), ketersediaan (*availability*), dan integritas (*integrity*). Tujuan dari serangan berbasis *web* secara signifikan berbeda serangan itu lainnya. Sebagian besar pengujian serangan tradisional untuk mempelajari jaringan atau *host* sebagai target serangan. Serangan berbasis *web* fokus pada aplikasi itu sendiri dan fungsi pada *layer 7* dari *OSI*. John Pescatore dari grup *Gartner* mengklaim bahwa hampir 70% dari semua serangan terjadi pada *layer* aplikasi (Crist, 2007).

Kerentanan pada aplikasi *web* bisa menyediakan sarana bagi seorang *hacker* untuk mengeksploitasi suatu sistem untuk melakukan tindakan kejahan seperti pencurian informasi atau perusakan sistem. Pencurian informasi yang dicuri dapat berupa identitas seseorang, berkas rahasia perusahaan, *lisensi software*. Target lain yang populer untuk serangan ini adalah data kartu kredit yang terlindungi dan tidak terenkripsi dapat digunakan untuk menyebabkan kerugian finansial untuk organisasi atau penggunanya.

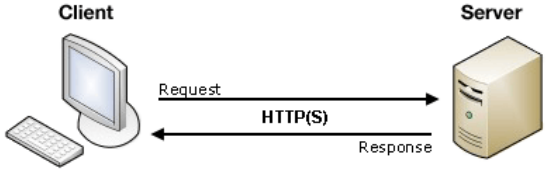
Bagaimana serangan kepada *web* itu terjadi? Jelas ada banyak metode dan teknik untuk melakukan penyerangan *website* ini. Berikut akan di jelaskan beberapa metode cara pendeteksian *web attack* melalui *access log* berdasarkan penelitian SANS Institute oleh Roger Meyer yang berjudul "*Detecting Attacks on Web Applications from Log Files*".

6. PHP, HTML dan *MySQL*

Secara sederhana script *HTML* diawali dengan *<html> <head> </head> <body>* kemudian diakhiri dengan *</body> </html>*, sedangkan *script PHP* diawali dengan *<?php* dan diakhiri dengan *?>*. Keduanya merupakan penyusun utama sebuah halaman *website* sedangan untuk ruang penyimpanan data yang besar atau informasi yang disediakan dalam sebuah website biasanya disimpan dalam sebuah *database*. Salah satunya *database* paling popular adalah *MySQL database* (Ikhsaan, Ilzamul. 2007). Berikut akan dijelaskan lebih lengkap tentang *PHP, HTML* dan *MySQL*.

7. Web Server

Web Server adalah *software server* yang menjadi tulang belakang dari *World Wide Web(WWW). Web Server* berkomunikasi dengan clientnya(web browser) mempunyai protokol sendiri yaitu *HTTP(HyperText Transfer Protocol)*. Dengan protokol ini komunikasi antar *web server* dan *client(browser)* dapat saling dimengerti dan lebih mudah (Abdullah, 2012).



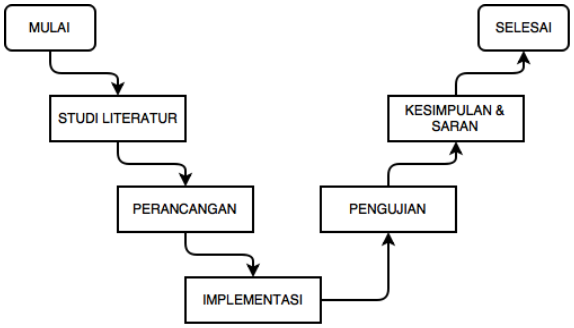
Gambar 2. 1 Komunikasi *Client Server*

Seperti pada gambar 2.3 proses dimulai dari permintaan *client (browser)*, diterima *web server*, diproses, dan dikembalikan hasil prosesnya oleh *web server* ke *web client* dan dilakukan secara transparan. Setiap orang dapat dengan mudah mengetahui apa yang terjadi pada tiap-tiap proses. Secara garis besarnya *web server* hanya memproses semua masukan yang diperolehnya dari *web client*-nya.
Berikut jenis-jenis *Webserver*:

1. *Apache Web Server*
2. *Apache Tomcat*
3. *Microsoft Windows server 2003 Internet Information Service (IIS)*
4. *Light HTTP*
5. *Sun Java System Web Server*
6. *Zerus Web Server*

8. Metodologi

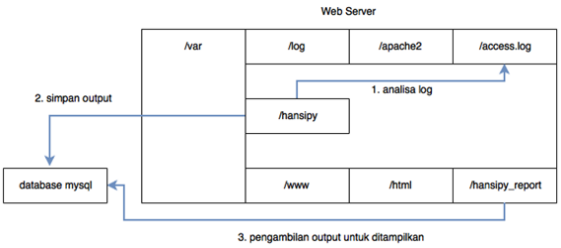
Langkah-langkah yang dilakukan dalam penelitian ini ditunjukkan pada gambar 3.1 dibawah ini.



Gambar 3.1 Diagram Alur Penelitian

9. Perancangan *Log Anayzer*

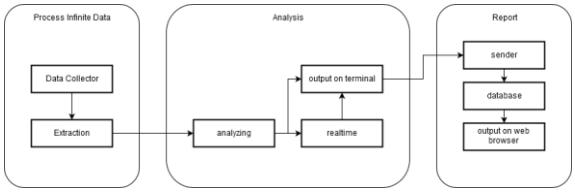
Perancangan *log analyzer* untuk mendeteksi serangan ini diletakkan pada suatu *directory* dalam *web server linux* dalam penelitian ini memakai linux *debian-based* yaitu *ubuntu*. Untuk semua modul dengan funsinya masing-masing yang telah telah dibuat dimasukkan satu *directory* dengan nama *hansipy*(nama *log analyzer*) dalam *directory* */var*. Tujuan disimpanya *log analyzer/hansipy* dalam folder */var* adalah untuk memudahkan sinkronisasi antara pengambilan file *input* yaitu *access log* yang umunya berada pada *path /var/log/apache2/* dan file output yang akan didistribusikan kedalam *web report* yang umumnya berada pada *path /var/www/html/*. Berikut blok diagram *log analyzer/hansipy* (gambar 4.1) mulai pengambilan data *input* sampai pengiriman hasil sebagai *output.*



Gambar 4.1 Konsep *Log Analyzer*

10. Perancangan Sistem *Log Analyzer/Hansipy*

Perancangan sistem *log analyzer/hansipy* ini bisa diartikan file apa saja yang digunakan dalam *directory hansipy* sehingga dalam pembahasan sebelumnya disebutkan bahwa *hansipy* dapat menganalisa input(*access log*) dari *path /var/log/apache2/* lalu menyimpan output dalam *database mysql* dan menampilkan dalam *website* lokal atau memanggil data yang terdapat pada *database mysql.* Dalam perancangan sistem *Log Analyzer* terbagi kedalam tiga bagian perancangan, yang pertama perancangan program untuk menganalisa log, lalu perancangan program untuk menjalankan program pertama secara *realtime* dan program penyimpan hasil kedalam *database mysql* dan yang terahir adalah rancangan program untuk pengambilan hasil/*output* dari program sebelumnya. Semua rancangan tersebut meliputi proses *data collector*, *extraction*, *analyzing and detecting*, *processing infinite data*, *sendout*, dan *reporting*. Berikut diagram alur proses sistem *analyzer* bekerja (gambar 4.2).



Gambar 4.2 Perancangan Sistem *Log Analyzer/Hansipy*

11. Implementasi Sistem

Implementasi adalah tahapan penerapan pada perangkat keras dan pengerjaan kode sesuai analisa dan perancangan pada bab sebelumnya. Dalam penelitian ini, penulis mengimplementasikan sistem *analyzer/hansipy* yang dirancang menggunakan bahasa pemrograman *Python* dan menampilkan *output* menggunakan bahasa pemrograman *PHP(Hypertext Preprocessor)* yang bersifat open source beserta penggunaan basis data yang melibatkan *DBMS(Database Management System) MySQL.*

Implementasi sistem berdasarkan hasil analisa sistem dan perancangan yang sudah dilakukan akan dibagi menjadi beberapa bagian seperti berikut :

1. Implementasi sistem *analyzer/hansipy* menggunakan *Python.*
2. Implementasi sistem *reporting result* menggunakan *PHP/HTML.*

12. Spesifikasi Perangkat Lunak dan Perangkat Keras

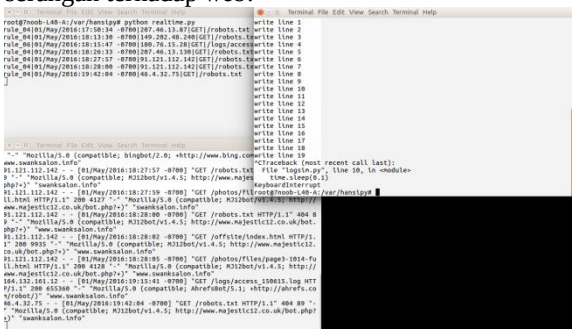
Perangkat lunak dan perangkat keras yang digunakan dalam pembuatan sistem *analyzer/hansipy* tersebut berpengaruh penting terhadap penelitian ini guna memperlancar pembuatan dan penggunaan sistem. Tabel 5.1 di bawah ini menunjukkan spesifikasi yang digunakan dalam penelitian :

Tabel 5.1 Keterangan Spesifikasi Perangkat

No.	Jenis Perangkat	Komponen
1	Perangkat Keras	a. Intel(R) Core(TM) i5-3337U
		b. RAM 4.00 GB
2	Perangkat Lunak	a. Ubuntu 16.04.1 desktop amd64
		b. Apache/2.4.18
		c. MYSQL 5.7.16-0ubuntu0.16.04.1
		d. Phpmyadmin 4.5.4.1deb2ubuntu2
		e. Python 2.7.12
		f. PHP 5.6.29-1+deb.sury.org~xenial+1
		g. Firefox 50.1.0/Chromium 55.0.2883.87

13. Implementasi Sistem *Analyzer/hansipy* menggunakan *Python*

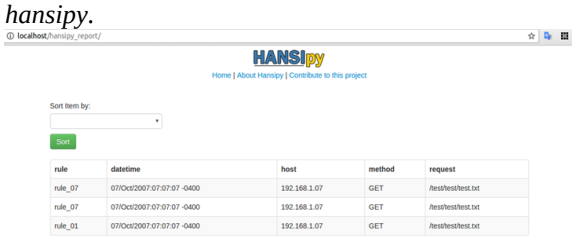
Seperti yang sudah dibahas sebelumnya pada bab perancangan sistem *log analyzer/hansipy* tedapat 3 langkah (gambar 4.1) yaitu analisa *log*, penyimpanan *output* dan pengambilan *output* untuk ditampilkan. Implementasi disini berhubungan dengan poin pertama dan kedua dengan menggunakan bahasa pemrograman *python* bertujuan untuk mendeteksi serangan dan menyimpan log yang berpotensial serangan kedalam *database*. Berikut hasil *capture desktop* ketika mendeteksi adanya serangan terhadap *web*.



Gambar 5.1 Capture Desktop *hansipy*

14. Implementasi Sistem *Reporting Result* menggunakan *PHP/HTML*

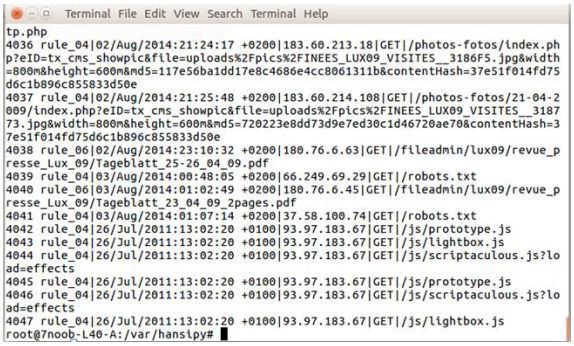
Berbeda dengan Implementasi sebelumnya pada sub-bab ini implementasi berfokus kepada poin ke-tiga yaitu pengambilan *output* untuk ditampilkan kembali dalam *web report hansipy*. Untuk pengimplementasiannya sendiri sistem ini menggunakan bahasa pemrograman *PHP/HTML* sederhana untuk menampilkan data dalam *database*. Berikut hasil *capture desktop*/tampilan *web report hansipy*.



Gambar 5.2 Tampilan *Web Report hansipy*

15. Pengujian Fungsional

Pengujian funsional bertujuan untuk mengetahui apakah sistem yang dibuat sudah sesuai dengan apa yang dibutuhkan. Pengujian ini meliputi penggunaan fungsi pada program utama(*hansipy)* dan ketepatan informasi hasil/*output* yang berikan sistem. Skenario pengujian dengan melakukan pengambilan sampel *access log* dari *website/webserver* yang tersedia pada mesin pencari kemudian diambil beberapa data dari output serta pembuktikan kebenarannya. Berikut beberapa gambar dan tabel pengujian *hansipy* :



Gambar 6.1 Pengujian *hansipy.py*

Dalam gambar 6.1 terdapat 5 file *access log* yang penulis ambil dari beberapa *website* secara acak (kata kunci pada mesin pencari *google* : *intext:index of intext:access.log*) dan salah satunya didapat dari *webserver* lokal. sebagai data untuk diuji coba menggunakan *hansipy*. Berikut deskripsi setiap *file access log :*

Tabel 6.1 Deskripsi *file access log*

Nama File	Banyak line	Sumber
access.log	1294	http://www.swanksalon.info/
access.log.1	36495	http://www.inees.org/
access.log.2.gz	204	http://www.amore-restaurant.co.uk/
access.log.4.gz	100	127.0.0.1 (lokal)
access.log.5.gz	1	http://www.swanksalon.info/
	38094	

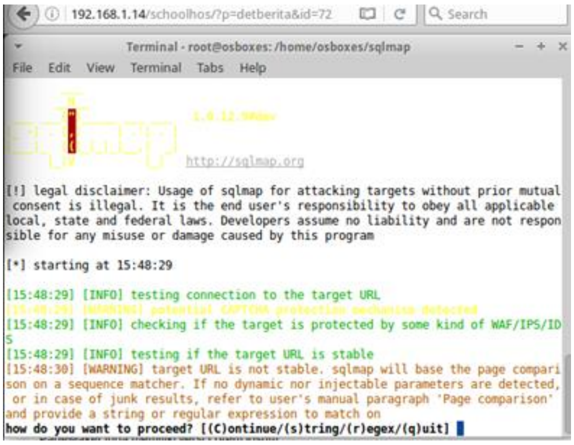
Pada gambar dan tabel 6.1 ditunjukkan bahwa sebanyak 5 *file access log* dengan 38094 baris/*line log* yang diuji mendapatkan 4047 baris *log* yang berpotensial dan dicurigai sebagai serangan terhadap *web*. Berikut beberapa baris *log* untuk setiap *rule* yang terdeteksi dalam program *hansipy* :

Tabel 6.2 Hasil/Result hansipy

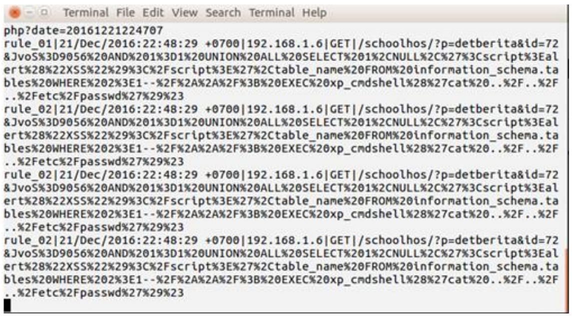
Kode Rule	Indikasi	Method	Request Url	Hasil Uji
rule_01	Cross-Site Scripting	GET	/http://<http//www.chaire.ecosoc.uqam.ca/portals/chaireecosoc/docs/pdf/bulletins/Bu	salah
rule_02	SQL Injection	GET	/index.php?option=com_ckforms&controller=ckdata&view=ckformsdata&layout=detail&task=detail&fid=-2+union+select+1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,group_concat(0x3C6B65793E,username,0x3a,password,0x3a,usertype,0x3a,block,0x3a,activation,0x3a,0x3C62723E,0x3C6B6579733E)UAH,35+from+jos_users+where+usertype='Super%20Administrator'+or+usertype='Administrator'%20	benar
rule_03	Insecure Direct Object Reference	GET	/system/fckeditor/editor/filemanager/browser/default/connectors/test.html	benar
rule_04	Page with status code between 400-500	GET	/robots.txt	benar
rule_05	Page with status code more than 500	GET	/sitemap.xml	benar
rule_06	Page login access	GET	/fileadmin/lux09/revue_presse_Lux_09/Tageblatt_25-26_04_09.pdf	benar
rule_07	Backdoor/Shell	GET	/twiki/cmd.php	benar

16. Pengujian *Realtime Mode*

Pengujian dangan *realtime mode* ini dilakukan dengan skenario penyerangan terhadap *server* lokal. *Server* lokal akan menjalankan *service website* dengan beberapa *cms* seperti *wordpress* dan *schoolhos*, sebelumnya juga sudah diketahui beberapa celah untuk menyerang website tersebut. Skenario pengujian dengan mengeksekusi *realtime.py* dengan target analiasa file *access log* dalam path */var/log/apche2*, kemudian dengan memanfaatkan fungsi *virtualbox* sebagai *virtual komputer* menjalankan *backbox*(*penetration test and security assessment*) untuk menyerang server lokal. Berikut salah satu gambaran skenario pengujian *realtime* dilakukan :



Gambar 6.2 *SQL Injection* pada Komputer Penyerang



Gambar 6.3 Deteksi *hansipy* Pada Sisi *Server*

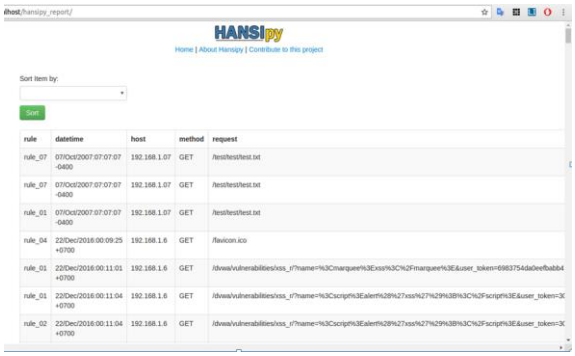
Dari serangkaian skenario serangan yang dilakukan ada beberapa penjelasan yang belum dijelaskan secara rinci pada sub-bab sebelumnya seperti penggunaan cms dan penjelasan hasil uji. Tabel berikut dapat menyimpulkan hasil percobaan tersebut.

Tabel 6.3 Kesimpulan Skenario Serangan

Skenario	Website	Penjelasan	Hasil Uji
Serangan XSS melalui browser	localhost/dvwa (vulnerable web application)	Mendeteksi adanya serangan reflected XSS. Untuk stored XSS tidak dapat dideteksi melalui access log.	sukses
SQL Injection menggunakan Sqlmap	localhost/schoolhos/?p=detberita&id=72 (schoolhos/Open Source CMS Sekolah)	Mendeteksi adanya serangan jenis SQL injection pada website	sukses
Scanning Menggunakan WPBrute	localhost/wordpress (CMS Wordpress)	Mendeteksi adanya serangan pada website	sukses
Scanning Menggunakan OWASP Zed	localhost/schoolhos	Mendeteksi adanya serangan pada website	sukses
Scanning Menggunakan Nikto	localhost	Mendeteksi adanya serangan pada website, namun hansipy terhenti karena kegagalan olah data	gagal
Shell/ backdooring	localhost/wordpress (CMS Wordpress)	Mendeteksi akses terhadap halaman login & mendeteksi adanya akses shell didalam website	sukses

17. Pengujian Pengolahan *Output*

Pengujian pengolahan output dilakukan untuk mengetahui fungsi *sender* atau pengirim hasil/*output* sudah benar atau tidak. Pengujian ini meliputi semua fungsi yang telah dilakukan sebelumnya menjadi kesatuan sistem yang dapat menganalisis *access log* secara *realtime* dan mengirimkan hasil ke database server lalu menampilkan dalam bentuk *web report*. Setelah dilakukan percobaan serangan pada sub-bab sebelumnya maka hasi/*result* hansipy akan dimasukkan dalam *database hansipy* dan ditampilkan kembali dalam *web report*. Berikut *screenshot* tampilan *web report hansipy* setelah dilakukan serangan.



Gambar 6.4 Tampilan *hansipy Web Report*

18. Kesimpulan

Berdasarkan pembahasan dari bab-bab sebelumnya, serta perancangan hingga pengujian kinerja sistem *analyzer/hansipy* yang telah dilakukan, maka dapat diambil beberapa kesimpulan sebagai berikut :

1. Berdasarkan pengujian fungsional dengan rata-rata hasil uji benar, maka bisa dikatakan sistem *log analyzer/hansipy* telah berhasil membaca log dan memilah antara *log* aman dan *log* yang berpotensi serangan.
2. Berdasarkan pengujian *realtime mode* dengan beberapa skenario dan rata-rata hasil uji sukses, maka bisa dikatakan sistem *log analyzer/hansipy* berhasil mendeteksi adanya serangan terhadap *website* secara *realtime*.
3. Berdasarkan pengujian pengolahan *output, web report* berhasil menampilkan data *log* yang berpotensi serangan dari *database* yang telah terisi data oleh proses pengujian sebelumnya, beserta fungsi navigasi dan fitur *sorting* berdasarkan rule yang berjalan dengan benar bisa dikatakan sistem *web report* berhasil terintegrasi dengan modul *hansipy*.

19. Saran

Penulis menyadari sistem yang telah dibuat masih banyak sekali kekurangan adapun saran yang dapat penulis berikan setelah penelitian ini yaitu untuk penelitian selanjutnya, terutama untuk masalah fitur dan pengkodean. Berikut beberapa saran yang bisa digunakan untuk penelitian selanjutnya.

1. Penambahan *pattern*(pola).
2. Pengembangan algoritma *signature–based* dimana *pattern* dapat menggunakan fungsi *regex*.
3. Penambahan sistem pendeteksian dengan algoritma *anomaly-based*.
4. Penambahan fitur *report* langsung kepada *device android* dengan aplikasi ataupun memanfaatkan *API telegram*.
5. Penambahan fitur *login web report hansipy*.
6. Pengembangan sistem *web report hansipy* menggunakan teknik *paging* agar *load* halaman *index* tidak terlalu lama.
7. Pengembangan *web report hansipy* menggunakan *css* sehingga tampilan lebih teratur dan mudah digunakan.
8. Penambahan fitur pada *web report hansipy* mengubah data menjadi bentuk bagan/grafik sehingga memundahkan admin memonitor *website*.

DAFTAR PUSTAKA

Academic Technology and Creative Services Spring. 2010. Web Design: An Introduction [ebook/pdf] tersedia di <http://www.mungi.org/wp-content/uploads/2015/10/Free-web-design-tutorial.pdf> > [diakses 27 Januari 2016]

Academic Technology and Creative Services. 2010. Web Design An Introduction [ebook/pdf] California State University. USA. tersedia di < www.csus.edu/indiv/s/snowdenr/WebDesign.pdf> [diakses 27 Januari 2016]

Beazley, D. 2008. Generator Tricks For Systems Programmers. [presentation slide/pdf] PyCon'2008. tersedia di <http://www.dabeaz.com/generators/Generators.pdf> [diakses 10 Oktober 2016]

Crist, J. 2007. Web Based Attacks. [paper/pdf] SANS Institute. tersedia di < https://www.sans.org/reading-room/whitepapers/application/web-based-attacks-2053> [diakses 11 Juli 2016]

Halterman, R. 2011. Learning to Program With Python. [ebook/pdf] Southern Adventist University. tersedia di < https://www.cs.uky.edu/~keen/115/Haltermanpythonbook.pdf> [diakses 20 Juni 2016]

Imperva. 2015. 2015 Web Application Attack Report (WAAR). [ebook/pdf] tersedia di < https://www.imperva.com/docs/HII_Web_Application_Attack_Report_Ed6.pdf> [diakses 11 Juli 2016]

Kevin, J. 2013. Sort Table Column in PHP/MySQL. [online] tersedia di < http://www.sourcecodester.com/php/5327/sort-table-column-phpmysql.html> [diakses 10 Oktober 2016]

Meyer, R. 2008. Detecting Attacks on Web Applications from Log Files. [paper/pdf] SANS Institute. tersedia di < https://www.sans.org/reading-room/whitepapers/logging/detecting-

attacks-web-applications-log-files-2074> [diakses 11 Juli 2016]

OWASP. 2013. OWASP Top 10-2013 The Top Ten Most Critical Web Application Security Risk. [paper/pdf] SANS Institute. tersedia di < https://www.owasp.org/images/f/f8/OWASP_Top_10_-_2013.pdf> [diakses 11 Juli 2016]

Sarokaari, N. 2012. How to identify malicious HTTP Requests. [paper/pdf] SANS Institute. tersedia di < https://www.sans.org/reading-room/whitepapers/detection/identify-malicious-http-requests-34067> [diakses 02 Agustus 2016]

Simmons, C. Ellis, C. Shiva, S. Dasgupta, D. Wu, Q. 2013. AVOIDIT: A Cyber Attack Taxonomy. [paper/pdf] University of Memphis. USA. tesedia di

<ais.cs.memphis.edu/files/papers/CyberAttackTaxonomy_IEEE_Mag.pdf> [diakses 26 Januari 2016]

Vijayarani, S. Sylviaa, S. 2015. INTRUSION DETECTION SYSTEM – A STUDY. [jurnal/pdf] Bharathiar University. tersedia di < http://airccse.org/journal/ijsptm/papers/4115ijsptm04.pdf> [diakses 10 Oktober 2016]

Walker, C. 2013 Introduction to the OWASP Mutillidae II Web Pen-Test Training Environment. [paper/pdf] SANS Institute. tersedia di < https://www.sans.org/reading-room/whitepapers/application/introduction-owasp-mutillidae-ii-web-pen-test-training-environment-34380> [diakses 02 Agustus 2016]

Wood, A. 2003. Intrusion Detection: Visualizing Attacks in IDS Data. [paper/pdf] SANS Institute tersedia di <w.visualinsights.com/markets/Intrusion%20Detection.pdf> [diakses 27 Januari 2016]

Zakharchenko, S. 2015 Scalp!/Anathema is a log analyzer for web server (Apache, nginx). [online] tersedia di <https://github.com/nanopony/apache-scalp> [diakses 10 Oktober 2016]