

## Anti-WebShell PHP Backdoor Scanner pada Linux Server

Christian Ronaldo Sopaheluwakan <sup>a,1\*</sup> dan Dian Widiyanto Chandra <sup>a,2</sup>

<sup>a</sup> Universitas Kristen Satya Wacana, Jl. Diponegoro 52-60, Salatiga, 50711, Indonesia

<sup>1</sup> 672016226@student.uksw.edu; <sup>2</sup> dian.chandra@uksw.edu

\* corresponding author

INFORMASI ARTIKEL	ABSTRAK
<p>Dikirim : 29 Juni 2020 Diulas : 23 Juli 2020 Direvisi : 30 Juli 2020 Diterbitkan : 27 Agustus 2020</p> <p><b>Kata Kunci:</b> Anti-Web Shell Deteksi Backdoor Backdoor Shell Keamanan Jaringan Server Linux</p>	<p>Backdoor atau yang biasa juga dikenal dengan istilah <i>web shell</i> merupakan salah satu kode yang digunakan <i>hacker</i> untuk <i>maintaining access</i> sistem secara ilegal. Beberapa program seperti <i>Anti Web-Shell</i>, <i>PHP Backdoor Scanner</i> yang bisa didapatkan secara gratis untuk menangani <i>web shell</i>. Namun program-program tersebut memiliki <i>database signature</i> yang tidak diperbaharui untuk menghadapi <i>PHP backdoor / Shell</i> saat ini. Sehingga diperlukan program <i>Anti Web-Shell</i> kontemporer yang mampu menghadapi <i>shell backdoor</i> saat ini. Penelitian ini menggunakan metode eksperimen dengan acuan penelitian serupa sebelumnya dan diimplementasikan langsung ke dunia industri profesional keamanan siber. Dengan memperkaya <i>signature</i> kamus <i>Regex</i> dan <i>String Array Matching Program Anti Web-Shell</i> yang sudah diaktualisasi dapat mendeteksi <i>backdoor</i> lebih banyak dibandingkan program serupa yang sudah lampau. Hasil dari penelitian ini adalah berupa <i>web application software</i> dalam ekstensi PHP. Aplikasi dapat meminimalisir 100% terjadinya <i>false positives</i> serta lebih cepat 2 kali lipat dalam memindai <i>files</i> karena lebih spesifik dalam melakukan <i>heuristic anlysis scan</i>.</p>
<p><b>Keywords:</b> Anti-Web Shell Backdoor Scanner Backdoor Shell Network Security Linux Server</p>	<p><b>ABSTRACT</b></p> <p>Backdoor or commonly also known as web shell is one of the malicious software that hackers use to maintain access systems that they have entered. Relatively few programs like Anti Web-Shell, PHP Backdoor Scanner circulating on the Internet, and can be obtained free of charge to deal with the issues above. But most of these programs have no actual database of signature behavior to deal with PHP backdoor / Shell nowadays. Then comes the contemporary Anti Web-Shell program that can deal with today's backdoor shell. This study uses an experimental method concerning previous similar studies and is implemented directly into the world of cyber security professional industries. By enriching the Regex dictionary signature and String Array Matching the actualized Anti Web-Shell program can detect more backdoor than similar programs that have existed in the past. The results of this study are in the form of a web application software in PHP extension. The application can minimize 100% of false positives and is twice as fast in scanning files because it is more specific in heuristic analysis scan.</p> <p>This is an open access article under the <a href="#">CC-BY-SA</a> license.</p> 

### I. Pendahuluan

Berdasarkan data dari *defaced websites* (zone-h.org), perhari ada belasan sampai puluhan situs pemerintah Indonesia (dengan domain \*.go.id) yang berhasil diretas. Proses peretasan ini tidak diketahui oleh pemilik situs tersebut bahkan dalam waktu yang sangat lama pengelola website tersebut tidak menyadari jika websitenya diretas[1]. Maraknya aksi “Deface” atau “Pembajakan” yang dilakukan para *Hacker* luar negeri dan juga dari dalam negeri terhadap situs-situs pemerintahan, organisasi, polisi, tentara, dan bahkan penyedia *provider* Internet besar Indonesia. Masalah seperti ini merupakan hal yang sangat miris dan patut dipertanyakan kredibilitasnya ketika *provider* layanan Internet di Indonesia bisa terkena aksi usil dari *Hacker* yaitu “Deface”. Hal ini menunjukkan bahwa kesadaran akan dunia IT *Security* di negara Indonesia masih rendah dan kurangnya SDM praktisi IT *Security* itu sendiri. Masalah ini merupakan hal yang perlu diperhatikan dan diutamakan ke depannya untuk pengelolaan website yang lebih baik khususnya dalam permasalahan *Cyber Security*.

“Deface” merupakan salah satu kegiatan merubah tampilan suatu *website* baik itu halaman utama, *index file*, atau pun halaman lain yang masih terikat dalam satu URL dengan *website* tersebut[2]. Kegiatan ini sendiri kerap disalahgunakan oleh para *Hacker* untuk tujuan bersenang-senang semata[3]. Aksi ini memerlukan yang

disebut *gaining access*[4],[5]. Pada umumnya *Hacker* memanipulasi *privileges access* untuk *gaining access* tersebut dengan cara menanamkan *backdoor access* yang ditanamkan dengan berbagai macam cara dan kreatifitas sang *Hacker*[6][7]. Faktanya sekarang ini *Backdoor* paling banyak dibuat dengan bahasa pemrograman PHP[8],[9]. Relatif sedikit *Softwares* serupa *Anti Web-Shell*, *PHP Backdoor Scanner* yang beredar di Internet & Bisa didapatkan secara *free* untuk menangani *issues* masalah diatas[10]. Tetapi kebanyakan dari *softwares* tersebut sudah tidak *up to date database behaviour signature*-nya untuk menghadapi *PHP backdoor / Shell* jaman sekarang[11].

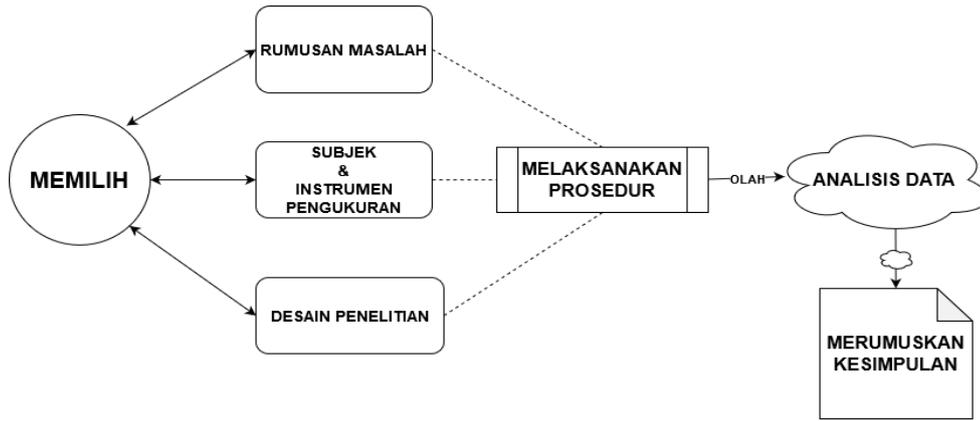
Penelitian terkait keamana *server* yaitu yang berjudul “Analisis Pendeteksian dan Pencegahan Serangan *Backdoor* Pada Layanan *Server*”. Penelitian ini menggunakan Metode Eksperimen, membahas tentang *backdoor* serta cara melakukan tindakan preventifnya. *Backdoor* adalah cara yang digunakan untuk masuk ke dalam sistem tanpa sepengetahuan *administrator*. *Backdoor* bertujuan untuk mempermudah memasuki sistem itu kembali jika jalan yang sudah dibuat dengan *exploit* telah ditutup oleh *administrator*. Maka dibuatlah simulasi yang saling berhubungan yang bertujuan untuk melakukan analisis terhadap *Backdoor* yang menggunakan IDS / IPS Snort [12]. Penelitian selanjutnya yaitu “Implementasi *Backdoor Scanner Tool* Menggunakan Metode *Carving File* Pada *Server* Codepolitan”, lebih ditujukan untuk *backdoor* dalam bentuk *source code* PHP dan *Image* yang memiliki ekstensi *jpg*, *png* menggunakan metode *carving file*, pendekatan *string* dan REGEX. Adapun tujuan penelitian ini dilakukan untuk merancang sebuah *tools* pemindai yang dapat mempermudah admin menemukan *backdoor* di dalam sebuah *server*[10].

Penelitian selanjutnya yaitu “*Basic Static Code Analysis* untuk Mendeteksi *Backdoor Shell* pada *Web Server*”, lebih fokus dalam scenario pengujian deteksi menggunakan *system* yang dibangun & Skenario pengujian deteksi menggunakan *PHP Shell Detector*[9]. Penelitian selanjutnya yaitu “Sistem Keamanan Jaringan Mendeteksi *Backdoor* Untuk Menemukan Celah Dan *Exploits* Pada *Web Server* Menggunakan Teknik IDS (*Intrusion Detection System*)”, penelitian ini membuktikan bahwa Teknik IDS (*Intrusion Detection System*) dapat digunakan untuk membantu proses deteksi file *backdoor* pada *web server* untuk menemukan celah dan *exploits*. Dengan teknik IDS berbasis *signature*, setiap *file* yang dideteksi akan dicocokkan dengan data *signature*[13]. Adapun penelitian lain yang berjudul “Analisis dan Deteksi *Malware* Menggunakan Metode *Malware* Analisis Dinamis dan *Malware* Analisis Statis”. Penelitian ini menjelaskan bahwasannya : Ada dua tipe analisis dalam melakukan analisis pada *malware* yaitu dengan analisis statis (analisa kode) dan analisis dinamis [8].

Berdasarkan penelitian-penelitian sebelumnya yang relevan dengan *backdoor* salah satu cabang substansi dari *Cyber Security* dapat disimpulkan bahwa *Backdoor* selalu berkembang. Pergerakannya dinamis, layaknya pembuat *Virus & Anti-Virus*. Jika ada *Virus* baru yang *release* maka harus dibuat serum yang baru sebagai penangkalnya. Maka dari itu penelitian ini hadir untuk menjawab hal tersebut. Perbedaannya adalah penelitian ini menganalisa versi *software* anti *backdoor* sejenis sebelumnya, mengkolaborasikan teori penelitian serupa sebelumnya untuk menyempurnakannya menjadi versi baru yang lebih Mutakhir, Optimal, Efektif, & Efisien, serta fitur yang lebih spesifik. Agar tidak hanya menjadi kesan omong-kosong belaka, penelitian ini nantinya akan diimplementasikan langsung di LAB PT. Datacomm Diangraha yang menggunakan *enviroment* Linux *Server* yang mendukung *Platform* PHP *Server*. *Software* *Anti Web-Shell*, *PHP Backdoor Scanner* yang berjalan pada *Platform* PHP *Server* yang pada umumnya digunakan untuk Linux *Server* ini akan dibuat lebih sempurna serta aktual dengan menyesuaikan riset *survey* berbagai macam *backdoor* yang beredar di Internet *recently*. Penelitian ini diharapkan akan membantu *SysAdmin Server / Administrator Website* dalam melakukan *maintaining scanning* rutin demi prosedur *security assessment* mereka. Dan apabila menggunakan *Software* *Anti Web-Shell* (AWS) yang sudah dirombak, diperbaharui, serta diberi bumbu *improvement* tambahan dari hasil penelitian yang menyesuaikan dengan jaman sekarang ini, harapannya membuat presentase angka keamanan pemilik atau pengelola *website* serta *Server* relatif semakin tinggi kiblbat tingkat keamanannya dari serangan penanaman PHP *backdoor* oleh *Hacker / Attacker* jaman sekarang.

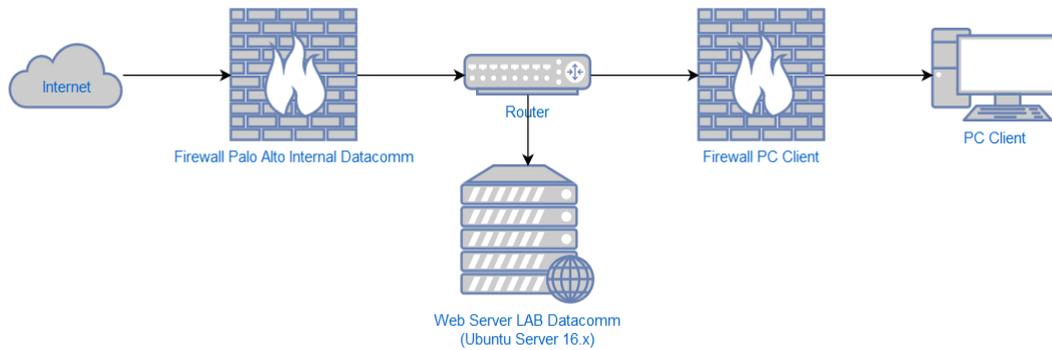
## II. Metode

Dalam penelitian ini penulis menerapkan Metode Eksperimen. Metode Eksperimen merupakan metode yang paling banyak dipilih dan paling produktif dalam penelitian. Apabila dilakukan dengan baik, studi eksperimental menghasilkan bukti yang paling benar terkait dengan hubungan sebab-akibat[12]. Menurut Gay (1981) dalam Emzir (2013:63) menyatakan bahwa penelitian eksperimental merupakan satu-satunya metode penelitian yang dapat menguji secara benar hipotesis menyangkut hubungan kasual (sebab-akibat). Langkah-langkah metode eksperimen dapat dilihat pada Gambar 1.



Gambar 1. Flowchart Pengimplikasian Metode Eksperimen

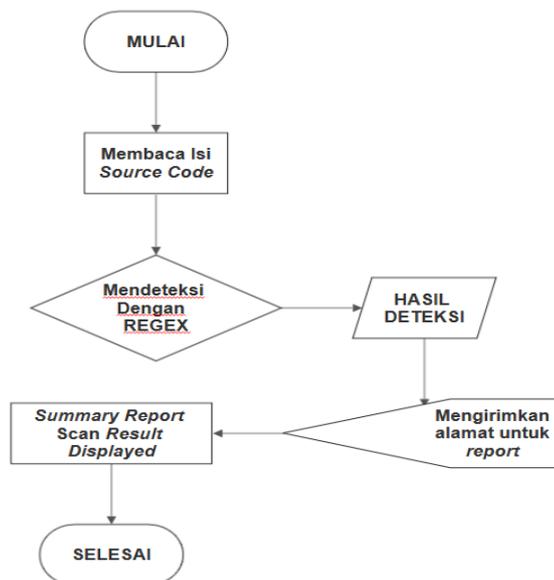
Berikut *design* topologi simulasi pada *internal* LAB Datacomm dapat dilihat pada Gambar 2.



Gambar 2. Web Server Machine LAB Datacomm

### III. Hasil dan Pembahasan

Flow Chart Software Lampau dapat dilihat pada Gambar 1.



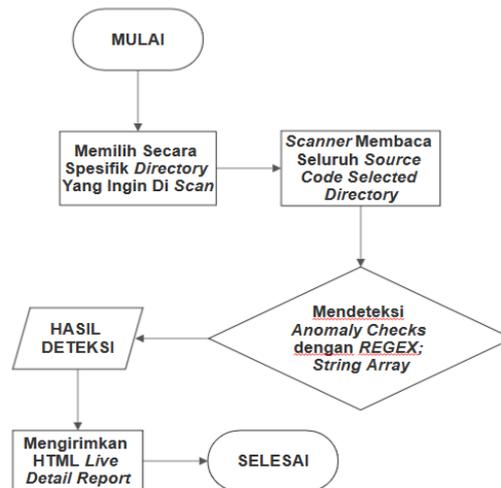
Gambar 3. Flow Chart Software Lampau

### Analisa Software Lampau:

Pada *software* lampau, modul *scanning* mengambil masukan *data* berupa *files* yang terdapat pada semua *directory* membaca *source code* masing-masing files secara membabi buta (*massive scan*), hal ini dapat memberatkan *server*[15]. Dilanjutkan dengan pencocokan PHP *Regex* yang relatif tidak di *update keyword signature* untuk *backdoors* dengan kondisi aktual di jaman yang sekarang ini. Hasil deteksi akan muncul, tentu saja presentase *false positive* relatif tinggi, karena kamus *Regex* yang tidak di *update* tersebut sangat minim (Bisa dilihat *highlight* kuning pada *Source Code* diatas). Lalu *Software* lampau ini akan mengirimkan alamat untuk *report* yang akan disuguhkan dalam bentuk *.html* sederhana (*result-scanner-lawas.html*). Kemudian jika dibuka akan menampilkan ringkasan hasil pemindaian berkas-berkas yang kemungkinan *backdoor infected* atau *safe*. Namun di *header title output scan result* meninggalkan sebuah pesan yaitu “*Be Carefull w/ False Positive*”, yang intepretasinya yakni pengguna di wanti-wanti untuk lebih menyaring lagi secara manual dan jangan percaya begitu saja dengan *Software* ini. Walaupun *Software* ini bersifat *passive preventive*, tetapi ia tidak lancang untuk melakukan keputusan *Modify* atau *Delete*. Yang mana hal ini akan dikembalikan oleh keputusan sang *User*.

### Penyempurnaan Software Anti-WebShell:

Belajar dari analisa di atas, intisari masalahnya adalah menambahkan kamus *Regex* yang lebih *up to date* untuk *signature backdoors* pada jaman sekarang. Tentunya dengan ikut serta mengkombinasikan beberapa metode selain *Regex* yaitu; *Anomaly Checks Techniques; Whitelisting & Blacklisting Techniques*[16]. Akan dikemas dengan menggunakan bahasa PHP, *Based GUI (Grapical User Interface)*. Memberikan kesan *More User Friendly* pada pengguna *software* kelak. Serta akan dibuat *compitable* dengan seluruh versi distro Linux. *Flow Chart Software Anti-WebShell* dapat dilihat pada Gambar 4.



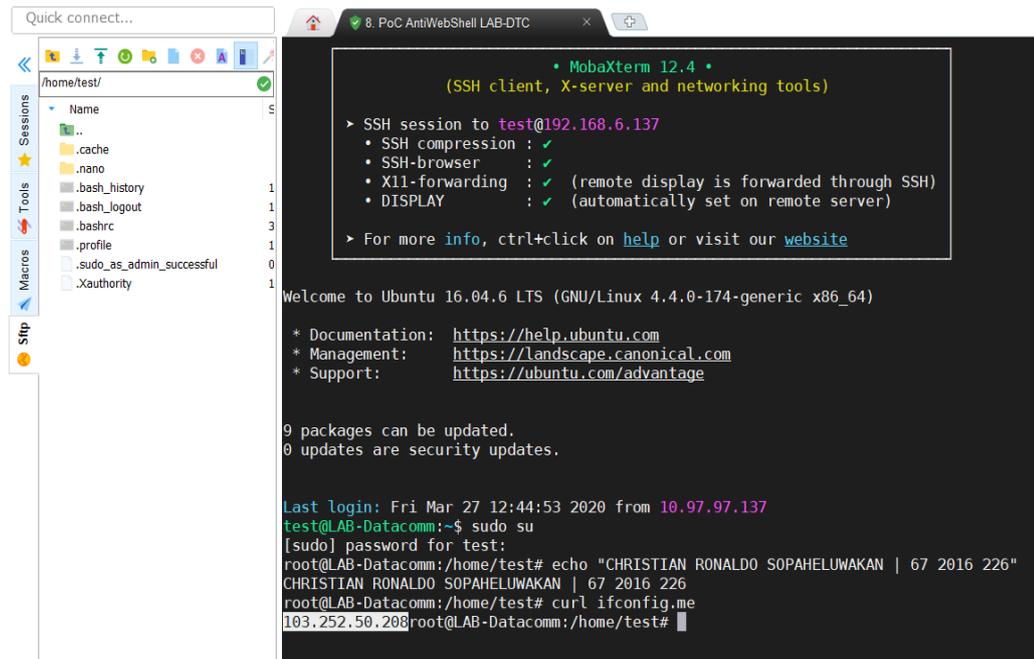
Gambar 4. Flow Chart Software AWS

Komparasi *Software* Lampau dengan *Software Anti-WebShell* yang telah dibuat akan diimplementasikan pada salah satu *server* LAB PT Datacomm Diangraha, yang menggunakan sistem operasi Ubuntu *Server* 16.04.6 LTS. Diakses secara *local* (DMZ) tetapi *Integrated* dengan IP WAN Datacomm yakni 103.252.50.200

Lookup IP Address

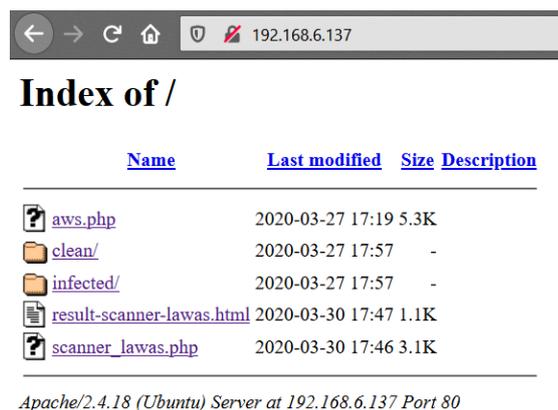
**Details for 103.252.50.200**

IP: 103.252.50.200  
 Decimal: 1744581320  
 Hostname: 103.252.50.200  
 ASN: 59134  
 ISP: PT. Datacomm Diangraha  
 Organization: PT. Datacomm Diangraha  
 Services: None detected  
 Type: [Broadband](#)  
 Assignment: [Likely Static IP](#)  
 Blacklist: Click to Check Blacklist Status  
 Continent: Asia  
 Country: Indonesia 🇮🇩  
 State/Region: Jakarta  
 City: Jakarta  
 Latitude: -6.1741 (6° 10' 26.76" S)  
 Longitude: 106.8296 (106° 49' 46.56" E)



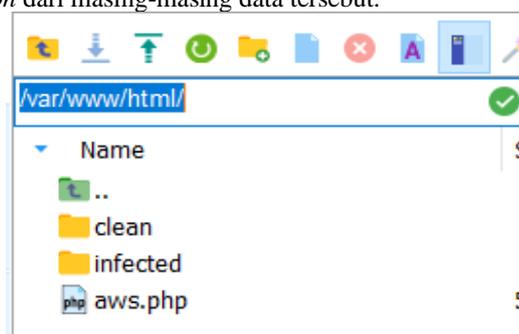
Gambar 5. Server LAB Datacomm

Jika melihat Gambar 5. Server LAB Datacomm maka terdapat *details* SSH Sessions to [test@192.168.6.137](https://192.168.6.137) dilanjutkan dengan akses IP tersebut secara lokal di browser.



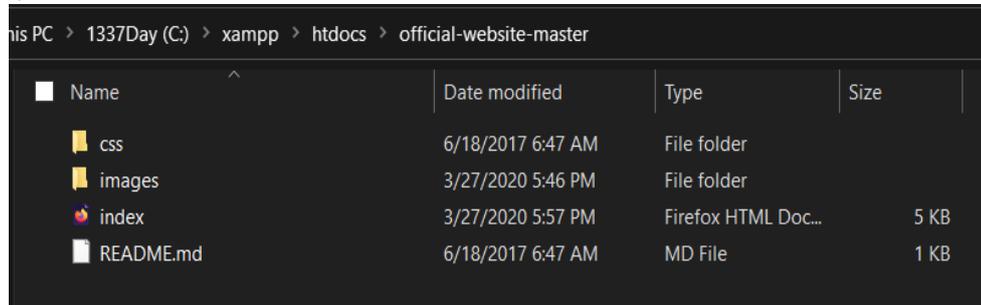
Gambar 6. Index of Server

Gambar 6 merupakan tampilan *Public Index Of Server* yang berjalan di Web Server salah satu Machine LAB Datacomm yang mana bisa dilihat Apache/2.4.18 menunjukkan Software Web Server yang digunakan, 192.168.6.137 merupakan IP Local dari salah satu Machine LAB tersebut, dan port 80 adalah default Port pengoneksian ke Web Server. Selain itu terdapat isi root directory dari Web Server yang meliputi files; aws.php, scanner\_lawas.php, result-scanner-lawas.html serta Folders /clean/ serta /infected/ dan bisa dilihat Last Modified, Size, serta Description dari masing-masing data tersebut.



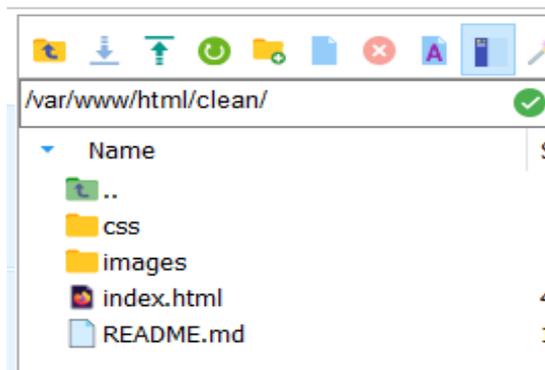
Gambar 7. Root Directory SysAdmin Perspective

Gambar 7 diambil dari sisi SysAdmin yang sedang *manage server*. `/var/www/html/` merupakan *Root Directory Default Web Server Apache*. SysAdmin membuat *folder /clean/* untuk simulasi suatu *directory* yang memang *clean & Safe* tanpa *backdoor infected*. Tetapi sebaliknya *folder /infected/* merupakan simulasi suatu *directory* yang sudah terinfeksi *backdoor & Ancaman* lain yang memungkinkan untuk *harming the system of the server*. Serta *Software Anti-WebShell* yang mana adalah *aws.php* telah dibuat atau diunggah ke dalam Apache Web Server.



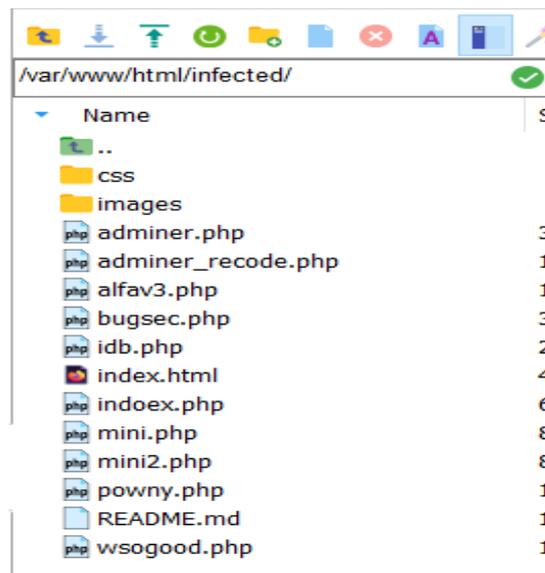
Gambar 8. Source Code Sample Portal Website

Gambar 8 merupakan *source code* dari portal *website* sederhana yang dibuat HTML5, *full static*. Dalam *folder /clean/ & /infected/* akan diunggah *source code* portal *websites* ini sebagai *sample* simulasi penelitian.



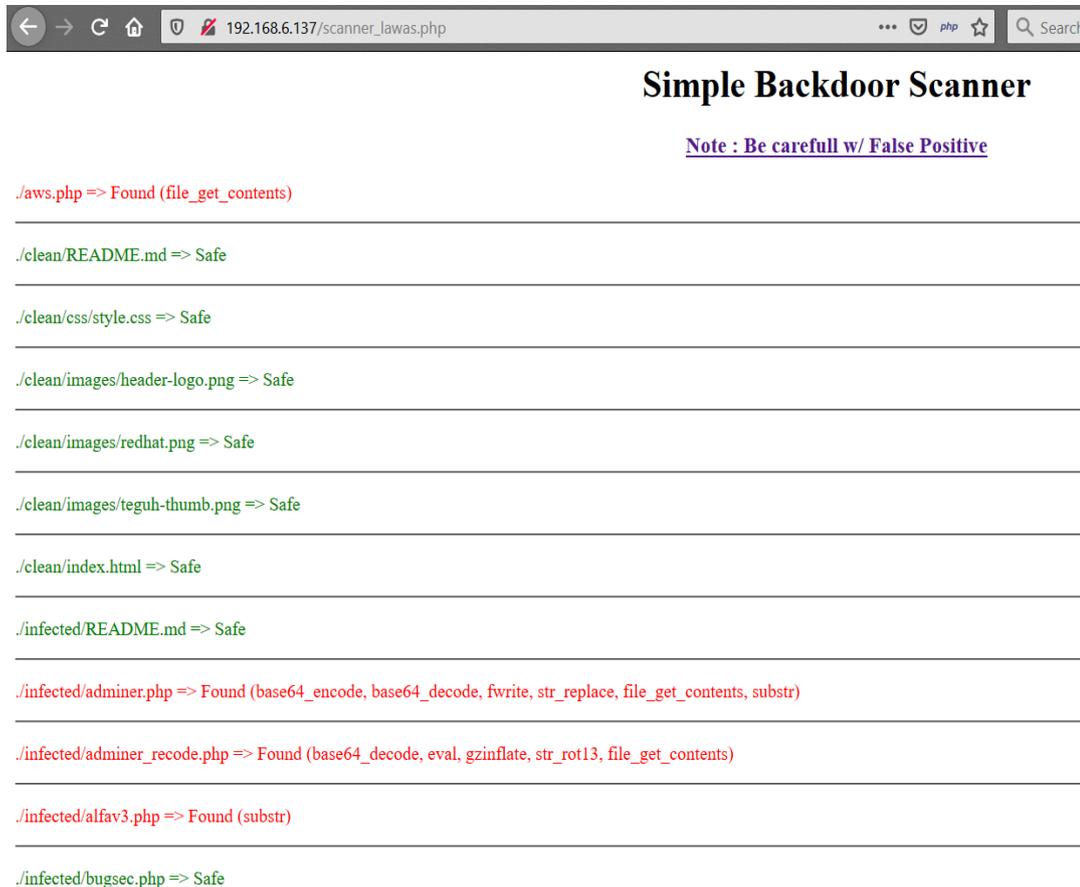
Gambar 9. Clean Folder

Gambar 9 menampilkan *folder /clean/* ini merupakan *folder* yang belum terinfeksi *backdoor*. Semua *Source Code Original* dari portal *website* sederhana yang telah dibuat di atas diunggah ke `/var/www/html/clean/`.



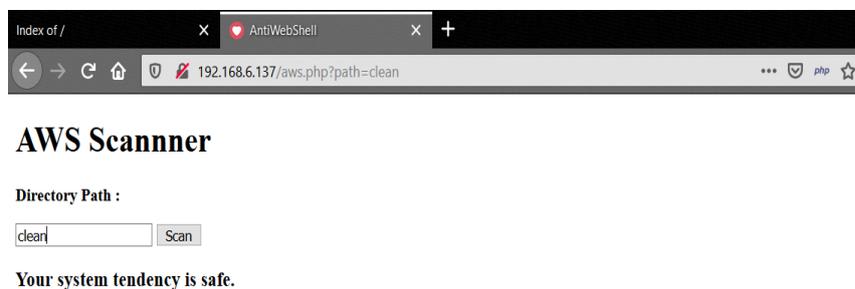
Gambar 10. Infected Folder

Gambar 10 menampilkan folder */infected/* ini merupakan *folder* yang sudah terinfeksi dengan banyak *type backdoors*. *Source code original* dari portal *website* sederhana yang dibuat sudah terkontaminasi dengan berbagai macam 10 *type backdoors* yang tertanam di */var/www/html/infected/* yakni; *adminer.php*, *adminer\_recode.php*, *alfav3.php*, *bugsec.php*, *idb.php*, *indoex.php*, *mini.php*, *mini2.php*, *powny.php*, *wsogood.php* yang mana *sample backdoors* ini di ambil dari *Internet*, *Komunitas Cyber Underground*, maupun ada beberapa *sample backdoor hasil modifikasi* yang diambil langsung dari praktisi *Cyber Security*.



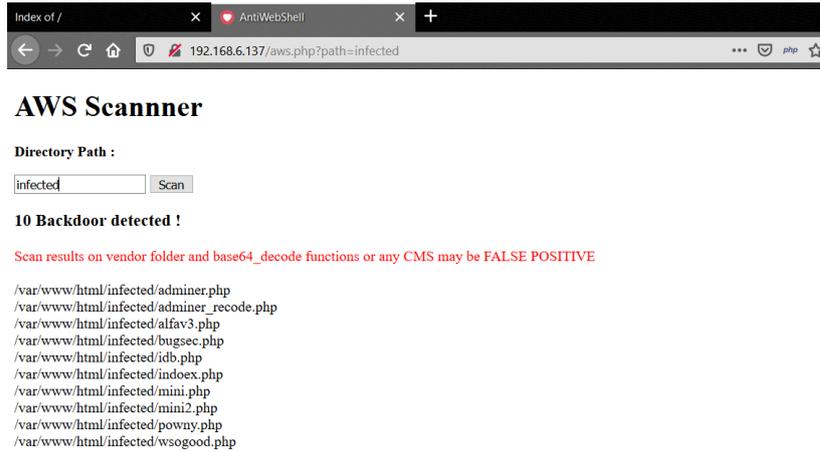
Gambar 11. *Scan Result with Old Software*

*Software* lampau sejenis (*Simple Backdoor Scanner*) yang didapatkan secara *free* di *Internet* melakukan *scanning* secara membabi buta (*massive scan*) tidak spesifik. Dan alhasil hanya mampu memprediksi ancaman 4 *backdoors* saja yang ditampilkan pada Gambar 11, yakni; *aws.php*, *adminer.php*, *adminer\_recode.php*, *alfa3.php* dan terdapat *note* bahwa masih banyak presentase kemungkinan *False Positive*.



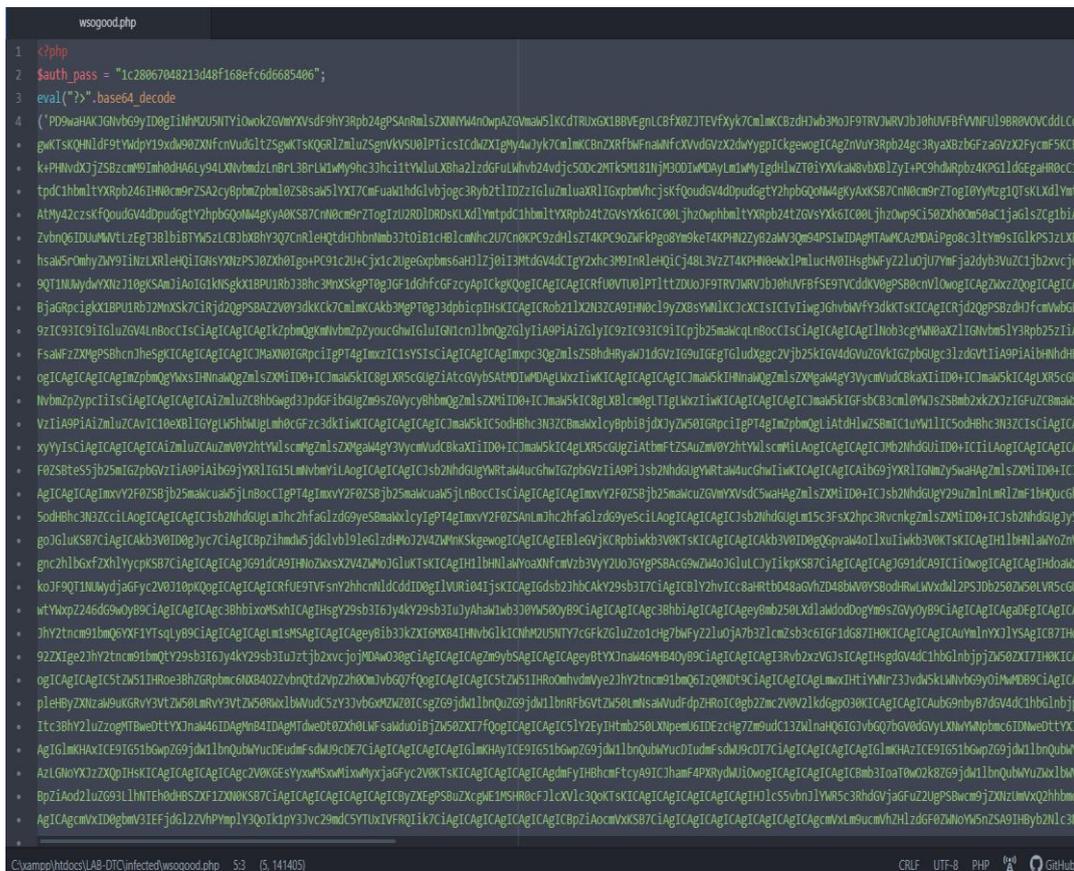
Gambar 12. *Scan /clean/ Folder with AWS Software*

Gambar 12 menampilkan *software Anti-WebShell (AWS)* melakukan *scanning* secara spesifik ke *Directory Path /clean/* dan *AWS system result* mencoba menyatakan prediksi dari *Software AWS* ini adalah “*Your system tendency is safe*”. Deklarasi tersebut menyatakan tendensi dari *folder /clean/* berdasarkan presentase aman dari infeksi *backdoor*. Karena sesuai skenario awal yakni folder */clean/* merupakan *original source code portal website HTML5, Full Static*, tanpa terkontaminasi *backdoors* yang tertanam.



Gambar 13. Scan /infected/ Folder with AWS Software

Gambar 13 menampilkan *software Anti-WebShell (AWS)* melakukan *scanning* secara spesifik ke *Directory Path /infected/* dan *AWS system result* menyatakan “10 Backdoor detected!”, 10 *type backdoors* ini adalah semua *backdoors* yang sudah disebutkan di skenario sebelumnya yakni; *adminer.php*, *adminer\_recode.php*, *alfav3.php*, *bugsec.php*, *idb.php*, *indoex.php*, *mini.php*, *mini2.php*, *powny.php*, *wsogood.php* yang semua berada di */var/www/html/infected/* sesuai skenario, *directory /infected/* ini memang berisi *original source code portal website* sederhana dari *HTML5*, *Full Static* akan tetapi kondisinya terkontaminasi oleh berbagai macam *backdoors* tertanam. Selain itu terdapat *alert* berupa “Scan results on vendor folder and base64\_decode functions or any CMS may be FALSE POSITIVE” yang merupakan usaha dari *Software AWS* mengingatkan akan gejala *False Positive* yang disebabkan oleh beberapa fungsi yang dianggap mencurigakan dari *Content Management System (CMS)* serta fungsi-fungsi mencurigakan lain seperti *base64\_decode*.



Gambar 14. Source code wsogood.php

Gambar 14 menunjukkan contoh *Source Code* dari salah satu *backdoors* pada simulasi yang telah dilakukan, yaitu *wsogood.php* dengan *full base64\_encode*. Rata-rata *signature* yang ditunjukkan *behaviour backdoors* yakni selalu menggunakan *base64\_decode*. Maka dari itu mudah saja bagi AWS untuk memindai dan *declare file wsogood.php* sebagai *backdoor*.

Status	Method	Domain	File	Cause	Type	Transferred	Size
200	GET	localhost:8080	scanner_lawas.php	document	html	2.29 KB	2.01 KB
404	GET	localhost:8080	favicon.ico	img	html	cached	1.03 KB

2 requests 3.04 KB / 2.29 KB transferred Finish: 26.04 s DOMContentLoaded: 25.71 s load: 25.74 s

Gambar 15. Waktu Scan Software Lampau

Gambar 15 menunjukkan aktifitas *realtime software* lampau mengeksekusi pemindaian *massive scan* secara langsung pada *root directory* dan melakukan 2 *request* dengan *method GET* file yaitu: *scanner\_lawas.php* & *favicon.ico* yang masing-masing tipenya adalah *document* dan *img*. Waktu *load* yang dibutuhkan sampai *scanner* benar-benar selesai bekerja adalah 25.74 s dan *DOMContentLoaded* 25.71 s.

Status	Method	Domain	File	Cause	Type	Transferred
200	GET	localhost:8080	aws.php?path=clean	document	html	868 B
200	GET	img.icons8.com	hearts.png	img	png	cached

2 requests 19.18 KB / 868 B transferred Finish: 274 ms DOMContentLoaded: 238 ms load: 282 ms

Gambar 16. Waktu Scan AWS @clean

Gambar 16 menunjukkan aktifitas *realtime software* AWS mengeksekusi pemindaian secara spesifik pada *folder /clean/* dan melakukan 2 *request* dengan *method GET* file yaitu: *aws.php?path=clean* & *hearts.png* yang masing-masing tipenya adalah *document* dan *img*. Waktu *load* yang dibutuhkan sampai *scanner* benar-benar selesai memindai *folder /clean/* hanya 282 ms dan *DOMContentLoaded* 238 ms.

Status	Method	Domain	File	Cause	Type	Transferred
200	GET	localhost:8080	aws.php?path=infected	document	html	1.43 KB
200	GET	img.icons8.com	hearts.png	img	png	cached

2 requests 19.76 KB / 1.43 KB transferred Finish: 635 ms DOMContentLoaded: 255 ms load: 303 ms

Gambar 17. Waktu Scan AWS @infected

Gambar 17 menunjukkan aktifitas *realtime software* AWS yang sedang mengeksekusi pemindaian secara spesifik pada *folder /infected/* dan melakukan 2 *request* dengan *method GET* file yaitu: *aws.php?path=clean* & *hearts.png* yang masing-masing tipenya adalah *document* & *img*. Waktu *load* yang dibutuhkan sampai *scanner* benar-benar selesai memindai *folder /clean/* hanya 303 ms dan *DOMContentLoaded* 255 ms. Kesimpulan hasil analisis bisa di lihat pada Tabel 1.

Tabel 1. Perbandingan Kinerja

No	Programming Language	Software Name	Load Speed	Backdoor Detected
1	PHP (CLI)	Old, Simple Backdoor Scanner	25.74 s – 25.71 s	4

No	Programming Language	Software Name	Load Speed	Backdoor Detected
2	PHP (GUI)	New, AWS Scanner	282 ms – 303 ms	10

Dengan kondisi sampel kasus yang sama *software* lama No. 1 menunjukkan rata-rata hasil *finished scan* diangka 25.74 s – 25.71 s (*second*). Akan tetapi *software* baru No. 2 hanya membutuhkan waktu dua kali lipat lebih cepat yakni 282 ms – 303 ms (*millisecond*) karena pencarian *custom folder* yang bisa lebih spesifik dalam melakukan *scanning files*. Kemudian *backdoor infected* yang terdeteksi pada *software* lama adalah 4/10 (40%) sedangkan *software* baru, AWS Scanner adalah 10/10 (100%) dikarenakan pengimplikasian *heuristic analysis method* dengan memanfaatkan fungsi *grep findstr* serta kamus *Regex* berkaitan dengan *keyword backdoor* yang aktual.

#### IV. Kesimpulan dan Saran

Berdasarkan penelitian *Anti-WebShell PHP Backdoor Scanner* Pada *Linux Server*, dapat disimpulkan bahwa Sistem *Software* sejenis *anti backdoor* yang lampau (*Simple Backdoor Scanner*) ini sudah tidak *up to date database signature, method* dan kamusnya. Sistem *Software* AWS yang dibangun ini diperbaharui secara aktual serta dikemas lebih *User Friendly*, menyesuaikan perkembangan jaman *Cyber Security*, dengan menambah serta mengkombinasikan beberapa Teknik & Metode baru serta memperkaya *signature* kamus *Regex & String Array Matching* sehingga lebih optimal dan maksimal pemanfaatannya saat mencari *infected backdoors* pada *web server*. Adapun saran untuk penelitian serupa lebih lanjut yakni dapat menambahkan jenis *Backdoors* yang menggunakan bahasa pemrograman lain selain PHP, contohnya *ASP Backdoor*. Kombinasikan dengan metode *Text search Linux with grep or findstr Method* sebagai variasi penyempurna *scanner*. Kemudian dapat juga menambahkan jenis *file* lain untuk melakukan *scanning* yang lebih lengkap terhadap seluruh *files* yang terdapat pada *server*. Lakukan penelitian lanjutan dengan algoritma *machine learning* substansi dari *artificial intelligence* untuk mengenali *backdoors* dengan proses yang lebih cepat serta *full* automasi.

#### Ucapan Terima Kasih

Penulis mengucapkan terima kasih banyak kepada PT. Datacomm Diangraha yang telah memberi dukungan dengan menyediakan perangkat, tempat dan akses jaringan untuk penelitian ini.

#### Daftar Pustaka

- [1] T. S. Hartono, "Amankan Website Anda dari 'Defacement,'" 2011. <https://tekno.kompas.com/read/2011/12/02/16352968/Amankan.Website.Anda.dari.Defacement?page=all#page2> (accessed Apr. 13, 2019).
- [2] U. Ite and H. Pidana, "Kejahatan Defecting :," vol. 3, pp. 143–159, 2015.
- [3] C. Camilo, U. López, M. G. Peña, J. Luis, O. Quintero, and A. Estado, "Antidefacement - State of art," vol. 14, pp. 9–27, 2016, doi: 10.18046/syt.v14i39.2341.
- [4] B. Ghazali, M. Teknik, I. Universitas, and A. Yogyakarta, "Mendeteksi Kerentanan Keamanan Aplikasi Website Menggunakan Metode Owasp ( Open Web Application Security Project ) untuk Penilaian Risk Rating," pp. 264–275.
- [5] J. Javier, B. Andrade, and D. Gan, "A Forensics Investigation into Attacks on Linux Servers."
- [6] S. Kumar and D. Agarwal, "Hacking Attacks , Methods , Techniques And Their Protection Measures," vol. 4, no. 4, 2018.
- [7] P. H. P. W. Shell and G. Supriyatno, "Searching for Forensic Evidence in a Compromised Virtual Web Server against SQL Injection Attacks," vol. 12, no. 12, pp. 1057–1063, 2018.
- [8] T. A. Cahyanto, V. Wahangara, and D. Ramadan, "Analisis dan Deteksi Malware Menggunakan Metode Malware Analisis Dinamis dan Malware Analisis Statis," pp. 19–30.
- [9] N. I. Widiastuti and M. Iqbal, "Basic Static Code Analysis Untuk Mendeteksi Backdoor Shell Pada Web Server," *J. INFOTEL*, vol. 9, no. 2, p. 177, 2017, doi: 10.20895/infotel.v9i2.209.
- [10] T. Wijayanto, A. Susilo, T. Wijayanto, and A. Susilo, "Implementasi Backdoor Scanner Tool Menggunakan Metode Carving File Pada Server Codepolitan," pp. 141–148, 2017.
- [11] O. W. Purbo, *Keamanan Jaringan*. Jakarta, 2011.
- [12] M. Universitas, B. Darma, D. Universitas, B. Darma, J. A. Yani, and N. Plaju, "Analisis Pendeteksian dan Pencegahan Serangan Backdoor Pada Layanan," no. 12, pp. 1–10.
- [13] Ali Mahmudi, "Sistem keamanan jaringan mendeteksi backdoor untuk menemukan celah dan exploits pada web server menggunakan teknik IDS (Intrusion Detection System)," *Simki-Techsin*, vol. 01, no. 04, pp. 1–10, 2017, [Online]. Available: <http://simki.unpkediri.ac.id/detail/13.1.03.02.0003>.
- [14] H. Alnabulsi, "Textual Manipulation for SQL Injection Attacks," pp. 26–33, 2014, doi: 10.5815/ijcnis.2014.01.04.
- [15] J. Komputasi, "Pembangunan Sistem Operasi Berbasis Linux Menggunakan Metode Linux From

- 
- Scratch,” vol. 1, no. 2, pp. 30–37, 2014.
- [16] T. M. Aji, D. E. Riyanto, and H. A. Wibawa, “Penerapan web services dan regular expression untuk verifikasi alamat berbasis hasil penelusuran,” vol. 1, no. 1, pp. 38–51, 2012.