Desain dan Implementasi Deteksi WebShell Malicious Web Shell (Backdoor Trap)

Raditya Faisal Waliulu[a], Santrinita Trhessya Jumame[b]

[a] Program Teknik Rekayasa Komputer Jaringan, Politeknik Saint Paul, Sorong, Papua Barat
[b] Program Akuntansi Keuangan Publik, Politeknik Saint Paul, Sorong, Papua Barat

Naskah Diterima : 25 September 2020; Diterima Publikasi : 4 November 2020
DOI : 10.21456/vol10iss2pp1188-194

Abstract

We present a report on hacker attacks against production servers on increased PHP vulnerabilities through SQL Injection attacks, XSS (Cross Site-Scripting), Cookie hijack, miss configuration, social engineering, CSRF (cross site request forgery), OTP bypass (take over account) and others. Hacker attacks leave a backdoor or webshell that will be accessed remotely (remote), this is common in blackhat hackers. Provides a shelltrap framework to use for and perform and clean the backdoor on the server. Because the back door has characteristics, namely: (1) taking over the physical server or localrooting; (2) adaptation to the run time environment; (3) using global variables to access the server. Have evaluated shelltrap on realworld server tame PHP Script and PHP backdoor. The experimental results get high level detection results of 98 %.

Keywords: Web Security; Web Shells; Backdoor; Intrusion Detection; Probability Analysis: Security Linux

Abstrak

Kita menyajikan sebuah laporan mengenai penyerangan hacker terhadap server production pada vulnerable PHP meningkat melalui penyerangan SQL Injection, XSS (Cross Site-Scripting), Cookie hijack, miss configuration, social engineering, CSRF (cross site request forgery), bypass OTP (take over account) dan lain-lain. Penyerangan hacker meninggalkan sebuah backdoor atau webshell yang akan diakses jarak jauh (remote), hal ini lazim pada Hacker blackhat. Menghadirkan sebuah framework shelltrap berguna untuk mendeteksi dan membersihkan backdoor pada server. Karena backdoor memiliki karakteristik diantaranya : (1) mengambli alih server fisik atau localrooting;(2) adaptasi pada run time environment; (3) menggunakan variabel global mengakses server. Telah dievaluasi shelltrap pada server realworld benign PHP Script and backdoor PHP. Experimental results mendapatkan hasil high detection rate 98 %.

Katakunci: Web Security; Web Shells; Backdoor; Intrusion Detection; Analisis Kemungkinan; Keamanan Linux

1. Pendahuluan

Komputer forensik menjadi salah satu cabang ilmu dalam keamanan komputer. Penting sebagai investigasi untuk berbagai jenis kejahatan. Kejahatan digital dari penipuan situs online, penyebaran gambar yang mengganggu anak melalui internet, pelanggaran peretasan hingga pengambilan hak akses root.

Hak akses root merupakan sebuah spesial privileges dalam sebuah sistem operasi yang bertanggung jawab menginstall, mengkonfigurasi, serta pemeliharaan sebuah aplikasi yang berjalan. Pencapaian privileges tertinggi membutuhkan sebuah teknik local rooting yang menggunakan barisan script untuk masuk melalui celah kerentanan sistem operasi tersebut.

Menggunakan enkripsi mencegah akses ke file atau harddisk dengan menggunakan sebuah algoritma enkripsi, membuat data tidak dapat dibaca tanpa akses menggunakan kunci atau sandi. Enkripsipun tidak menjamin sebuah file atau hak akses jika tidak dilakukan update berkala dan menggunakan repository yang telah tersedia dari sistem operasi (menghindari pihak ketiga) (Simson, 2007).

Web shell atau biasa disebut backdoor merupakan sebuah script yang di unggah keserver web yang disusupi, bertujuan mengaktifkan akses presisten pada mesin yang disusupi. Backdoor lazim terdapat malware ataupun ransomware yang bertugas mematai dan menjadikan trigger korban melakukan aksi (Li *et al.*, 2019).

Pemindai malicious sofrware menggunakan signatures dan heuristik lain untuk mencari backdoor pada mesin server secara rekursif menggunakan algoritma tertentu. Skenario ini dikembangkan untuk mencegah dari pra-exploitasi dari serangan hacker untuk escalation privileges.

*) Penulis korespondensi: waliuluraditya@gmail.com

Fase pra-exploitasi mesin sistem operasi dilakukan berbagai aktivitas seperti SQL Injection, RFI (Remote File Inclusion), LFI (Local File Inclusion), Arbitrary File Upload, XSS (Cross-site Scripting). Setelah dilakukan salah satu dari celah ktivitas kemudian penyerang menyusupi backdoor shell. Hal ini sangat penting mengingat popularitas layanan web menjadi sangat besar dan kerentanan yang dilaporkan sangat tinggi.

Untuk mengatasi tantangan backdoor ini pada dunia internet. Kami telah merancang sistem detektor malicious software. Mempertimbangkan fakta bahwa PHP menjadi favorit bahasa pengembangan pemograman. Implementasi sistem deteksi malicious software terfokus pada PHP-based malicious backdoor.

Detektor backdoors mengevaluasi tiga elemen penting : (i) komunikasi user dan penyerang external, (ii) run time enviromment, (iii) dynamic signature. Secara nyata penelitian ini dibuat dengan mengikuti :

1. Mengusulkan sebuah fitur sintaks membedakan antara malicious script web dan php murni,
2. Kami memiliki validasi empiris menindaklanjutin detektor malicious script.
3. Kami mengevaluasi detektor malicious script dan php murni. Sebagai hasil percobaan.

Analisis malicious software dibedakan menjadi dua diantaranya analisis statis, dianalisis tanpa menjalankan malware tersebut. Sedangkan dinamis berbeda, harus dijalankan pada sistem atau lingkungan virtual agar tidak berdampak buruk pada sistem atau server production (Waliulu & Iskandar, 2018).

2. Kerangka Teori

Layanan Web adalah salah satu terpoopuler diantara internet services yang menawarkan layanan penting dan menyimpan data sensitif. Sebagai perangkat lunak dengan kerumitan yang sangat besar, banyak layanan web demikian sangatlah rentan (McDaniel & Rubin, 2005).

Upaya penelitian dikembangkan dan diinvestasikan untuk memecahkan tantangan keamanan web pada fase pra-exploitasi (yaitu, untuk mengidentifikasikan kerentanan dalam script web) (Barth *et al.*, 2009) (Wassermann & Su, 2008).

Mengusulkan kerangka kerja untuk menghasilkan kasus uji untuk mengungkapkan kerentanan sistem web. Memanfaatkan analisis program statis untuk mendeteksi aplikasi web PHP yang rentan terhadap injeksi SQL dan serangan XSS (Wassermann & Su, 2008) (Xie & Aiken, 2016). Kemudian dikembangkan metode identifikasi aplikasi web PHP dengan kerentanan semantik seperti loop tak terhingga dan tidak adanya pemeriksaan otorisasi (Son & Shmatikov, 2011).

Menjalankan malware disandbox pada virtual envionment merupakan salah satu langkah bijak dalam manajemen system resources (jaringan komunikasi dengan remote server dan menangani melalui pesan error) (Stiborek *et al.*, 2018).

Pada penelitian yang dilakukan oleh Zahra Salehi, menerapkan deteksi malware secara dinamis. setiap biner yang dijalankan dalam sandbox virtual environment terkontrol dan tidak mengganggu sistem utama yang sedang berjalan dalam sistem operasi. pendetensian ini malicious sofware menggunakan API yang setiap argumen (digital signature) direkam dan diproses pada backend (server-side) (Zahra *et al.*, 2017).

Mendeteksi malicious software ataupun backdoor pada penelitian Yujie Fan, adalah mangambil extract signature dari data backdoor yang diajukan keumudian menggunakan sebuah metode mining algorithm untuk menemukan pola malicious sequential patterns berikutnya di klasifikasikan menggunakan ANN (All Nearest Neighbor) untuk dikonstruksikan pada deteksi pola malware. hal ini efektif jika terdapat backdoor atau malicious terbaru (Yujie *et al.*, 2016).

2.1. Simple skrip backdoor PHP

Pada makalah ini, backdoor pada dasarnya sebagai fungsi pintu belakang untuk menyusupi server web. Setelah berhasil diunggah, penyerang dapat mengunjungi shell web miliknya server-side script, guna mengontrol server yang disusupi. Shell web akan menjalankan perintah ini dan menampilkan hasilnya ke penyerang.

Web shell dengan mudah bypassed htaccess yang telah dibuat oleh server untuk melindungi web services. Htaccess memiliki dampak besar terhadap server dan mempertahankan keberadaan perangkat lunak didalamnya. Namun, terkadang htaccess dan string keamanan luput.

Backdoor aka web shell dapat dibuat dengan bahasa pemograman apapun sesuai server target. Pada umumnya ditulis dalam bahasa PHP, ASP dan JavaScript. Backdoor PHP sederhana ditulis menggunakan PHP web based shell dengan cara yang paling sederhana memanfaatkan PHP built-in bernama “system ()”. Perintah eksekusi dilakukan pada url browser menggunakan parameter “cmd” meneruskan melalui terminal server, HTTP requests GET parameter kemudian backdoor sederhana ini disimpan dengan nama simple.php. Dapat dilihat pada Gambar 1.

```
1 <? php
2 system ($_GET['cmd']);
3 ?>
```

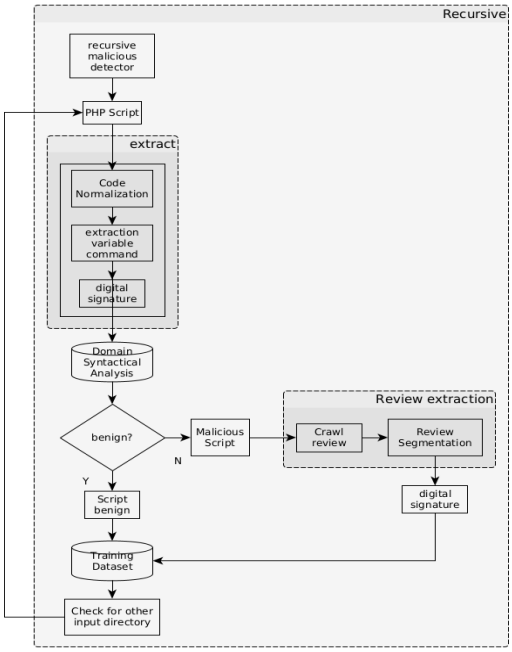
Gambar 1. simple PHP Backdoor

2.2. Sistem Deteksi

Dibuat sebuah sistem desain disebut trapshell untuk mendeteksi malicious web shell. Trapshell disusun berdasarkan dua fase diantaranya fase training

dan fase deteksi. Pada fase training pengklasifikasi statistik dilatih dari serangkaian script PHP benign dan berbahaya yang mengungkapkan teknik canggih yang diadopsi oleh penyerang untuk mencapai ketahanan dan kelincahan shell web, pada fase ini sebuah script dalam satu buah directory web server dipindai menggunakan fitur vector secara recurisive tiap sub/up directory kemudian dianalisis secara klasifikasi statis. Pada Gambar 2 mempresentasikan sebuah desain overview architectural dari trapshell.

Kemudian, badan script PHP diekstraksi kemudian dianalisis sehingga mengenerate sebuah digital signature untuk melatih Trapshell. Hasil dari digital signature disimpan dalam data latih untuk digunakan pada komparasi script PHP berikutnya. Pada fase detection, Trapshell mengidentifikasi sebuah benign script dan malicious script dengan melakukan komparasi menggunakan digital signature yang telah dilatih. Berikut arsitektur trapshell yang diajukan pada Gambar 2.



Gambar 2. Arsitektur Trapshell Deteksi Backdoor

2.3. Rekursif hash signature

Untuk menunjukkan keefektifan fitur yang kami buat, tidak ada upaya besar yang digunakan untuk menemukan rangkaian fitur yang optimal atau terbaik. Dengan menginvestasikan lebih banyak perhatian pada teknik pemilihan fitur pasti hasil yang lebih baik akan diperoleh. Pada tahap pertama, skor Fisher digunakan hanya untuk kecepatannya. Skor Fisher (Duda, Hard, & D.G, 2012) memilih fitur yang paling diskriminatif. Skor ini didefinisikan sebagai:

$$Fr = \frac{\sum_{i=1}^{c} n_i \, (u_i - u \;)^2}{\sum_{i=1}^{c} n_i \theta_i^2} \pi r^2 \quad (1)$$

n i adalah banyaknya sampel data di kelas i,
μ i adalah rata-rata nilai fitur di kelas i, deviasi standar dari nilai fitur dikelas i ditunjukkan oleh θ i, dan merupakan nilai fitur rata-rata secara keseluruhan Himpunan data.

2.4. Backdoor WebShell hacker

Web shell terenkripsi dilakukan oleh penyerang agar tidak setiap orang dapat menggunakan. Selain terdapat log yang terkirim penyerang, terpassword hanya pemilik yang berotentik menggunakan juga mencegah dari pencurian sourcecode dari recode script lamer mengubah copyright.

Web shell yang terkenal ada banyak bertebaran di forum underground beberapa disebutkan antara lain Fig.1: (i) WSO Web Shell by Orb; (ii) Marijuana Shell, (iii) B374K Shell dan (iv) alfashell. Setiap web shell memiliki perbedaan dan ciri khas menunjukkan fungsionalitas memudahkan penyerang bebas untuk menggunakan untuk aktivitas post-exploit, dapat dilihat pada Gambar 3.

3. Metode

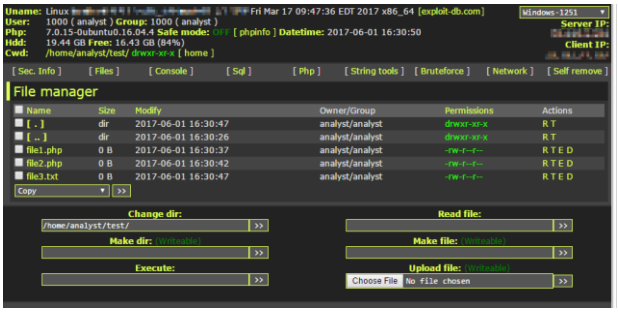
Fokus dari paper ini bertajuk pada trapshell dengan melatih secara syntactical and semantic menentukan antara malicious script PHP (backdoor) dan benign script PHP secara cepat, serta mengkarantina malicious script pada directory tertentu. Agar desain fitur mampu dengan efektif membedakan malicious script backdoor dari beningn script, kita mengidentifikasikan secara tiga karakteristik yang menjadi tipekal malicious web shell.

3.1. Penggunaan Variabel Global

Penggunaan variabel yang sering digunakan seperti $_GET, $_POST, $_COOKIE, $_FILES, $_SERVER, $_REQUEST, $_SESSION, $_ENV dan $GLOBALS. tiap variabel ini memiliki fungsi dalam parameter php script selain menjembatani komunikasi antara script php dengan mesin server juga sebagai pertukaran data dengan database.

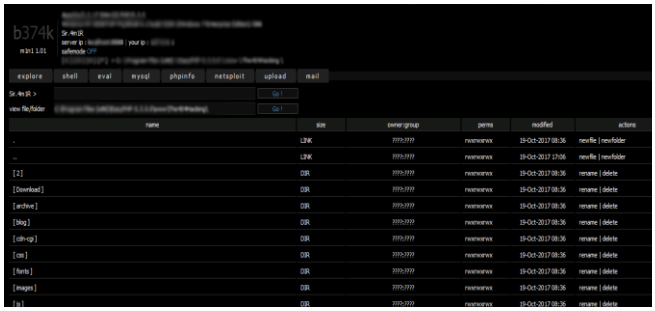
Variabel diatas sering digunakan untuk mengakses perintah dan parameter dari penyerang. Secara komparatif, tiap script beningn memiliki fungsi yang dilokalkan. Dimana, setiap script menangani variabel super global untuk memfasilitasi pengembangan dan debugging. Oleh karena itu, kami mengusulkan fitur berikut untuk mengukur pengamatan kami.

Masih banyak variabel menyajikan distribusi global dalam script PHP, script benign atau web backdoor sangat berbahaya. Seperti yang ditunjukkan dalam distribusi, perkiraan hampir 64% dari webshell backdoor berbahaya mengacu pada variabel super global selama lebih dari 20 kali sedangkan sebagian besar (98%) script beningn mengakses variabel super global kurang dari 20 kali.

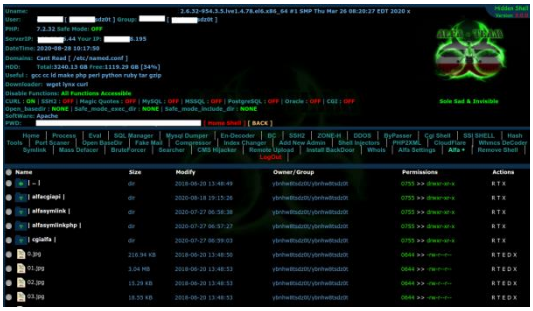


(i) WSO Web Shell

(ii) Marijuana Shell



(iii) B374K SHell



(iv) AlfaShell

Gambar 2. *Backdoor Populer WebShell*

3.2. Bypass Server-Side Validation

Diberikan sebuah script validasi upload gambar pada server-side untuk memvalidasi sebuah tipe gambar yang lazim di upload pada beberapa platform web aplikasi. Dapat dilihat pada Gambar 4.

```
<script type="text/javascript">
 var _validFileExtensions = [".jpg", ".jpeg", ".bmp", ".gif", ".png"];
 function Validate(oForm) {
 var arrInputs = oForm.getElementsByTagName("input");
 for (var i = 0; i < arrInputs.length; i++) {
 var oInput = arrInputs[i];
 if (oInput.type == "file") {
 var sFileName = oInput.value;
 if (sFileName.length > 0) {
 var blnValid = false;
 for (var j = 0; j < _validFileExtensions.length; j++) {
 var sCurExtension = _validFileExtensions[j];
 if (sFileName.substr(sFileName.length - sCurExtension.length, sCurExten
LowerCase() == sCurExtension.toLowerCase()) {
 blnValid = true;
 break;
 }
 }

 if (!blnValid) {
 alert("Sorry, " + sFileName + " is invalid, allowed extensions are: " +
 return false;
 }
 }
 }
 }

 return true;
 }
 </script>
```

Gambar 4. Validasi Skrip Tipe File Upload

Gambar 4, menjelaskan file JavaScript ini hanya memproses permintaan Anda sebelum benar-benar dikirim ke server dan memeriksa apakah file Anda memiliki ekstensi dari file gambar (jpg, jpeg, bmp, gif, png). Yang dapat dimanipulasi setelah Anda menghentikan permintaan dan tamper parameternya untuk mengubah konten dan nama file dari gambar yang diunggah dengan malicious code yang sebenarnya dan dengan ekstensi yang dapat dijalankan oleh penyerang.

3.3. Tipe Konten Validasi

Validasi Tipe Konten adalah ketika server memvalidasi konten file dengan memeriksa tipe MIME file, yang dapat ditampilkan di permintaan http. Misalnya, beberapa unggahan file gambar memvalidasi gambar yang diunggah dengan memeriksa apakah Jenis Konten file tersebut adalah jenis gambar. Beberapa sintaks php program dapat dilihat pada Gambar 5.

```
<?php
 $mimetype = mime_content_type($_FILES['file']['tmp_name']);
 if(in_array($mimetype, array('image/jpeg', 'image/gif', 'image/png'))) {
 move_uploaded_file($_FILES['file']['tmp_name'], '/uploads/' . $_FILES['file']['name']);
 echo 'OK';

 } else {
 echo 'Upload a real image';
 }
```

Gambar 5. PHP Skrip Validasi

Gambar 5, dapat dijelaskan dengan bagaimana cara bypass validasi tersebut. Hal ini merupakan jenis validasi yang dapat dilewati dengan mengubah nama file, misalnya menjadi "shell.php" atau "shell.aspx" tetapi mempertahankan parameter "Jenis Konten" sebagai Jenis Konten "image / *". Seperti "image / png", "image / jpeg", dan "image / gif".

Pada sintaks pengkodean Gambar 5, kode sebelumnya kita dapat melihat bahwa memeriksa tipe MIME yaitu ContentType dari file yang sedang diupload ke server, seperti yang ditunjukkan di atas dalam hal ini kode ini hanya menerima image / jpeg, image / gif, image / Jenis file png. Kita dapat dengan

mudah melewati jenis validasi ini dengan mengunggah file yang dapat dieksekusi tetapi setelah kita memanipulasi permintaan dan mengubah kolom "ContentType" menjadi jenis MIME dari gambar yang diterima server web.

4. Hasil dan Pembahasan

Tujuan penulis melakukan penelitian adalah untuk menentukan apakah backdoor dengan efektif dan effisien terdeteksi oleh shelltrap dan menciptakan sebuah framework server yang kebal terhadap backdoor hingga mendeteksi false positif.

4.1. Sample Host Uji Coba

Pada penelitian ini diberikan sample empat host server masing-masing dengan keterangan directory serta files didalamnya. Perlu di ingat bahwa terdapat subdirectory dalam dicrectory serta files dalam subdirectory. Sehingga akan diukur recursive, kecepatan deteksi dan memory yang dikonsumsi oleh shelltrap. Berikut Tabel 1 informasi sample uji coba shelltrap.

Tabel 1. Sample Host Uji Coba Backdoor

No	Host	Directory server	directory	files
1	Infokegiatan	/var/www/html/infokegiatan/public_html/	179	1085
2	bpbd	/var/www/html/bpbd/public_html/	355	2245
3	daerah	/var/www/html/daerah/public_html/	355	2234
4	econsul	/var/www/html/econsul/public_html/	399	2437
5	bapenda	/var/www/html/bapenda/public_html/	415	2664

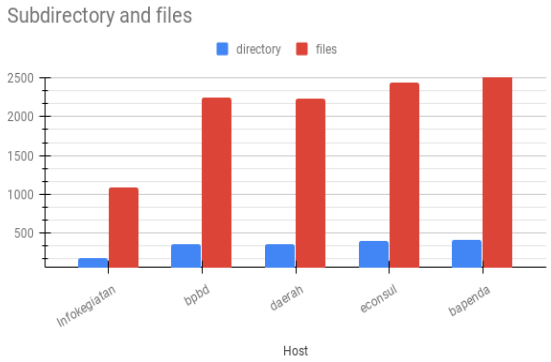
Tabel 1, merupakan sampel uji coba terhadap backdoor yang akan dimasukkan kedalam subdirectory pada 5 sample host. Akan ditaruh random bahkan bisa jadi terdapat backdoor dalam directory dan subdirectory. Hal ini dilakukan seperti real case pada advanced persistent threat. Akan diukur kecepatan recursive, kecepatan deteksi serta berapa memori yang dikonsumsi oleh shelltrap tersebut.

Dari data diatas, perhitungan akurasi Shelltrap yang diusulkan telah dievaluasi menggunakan rumus perhitungan seperti dibawah ini.

$$Accuracy = \frac{TP + FN}{(TP + TN + FP + FN)} \quad (1)$$

Dimana, TP = True Positif; TN = True Negatif; FP = False Posifit; FN = False Negatif.

Tabel 1, divisualisasikan Pada Gambar 6 menyangkut subdirectory dan jumlah files. Pada umumnya tiap folder dalam sebuah webapps tergantung dari fungsi. Beberapa diantaranyanya sebagai view, responsive screen, backend, database dan bahkan sebagai javascript serta css.



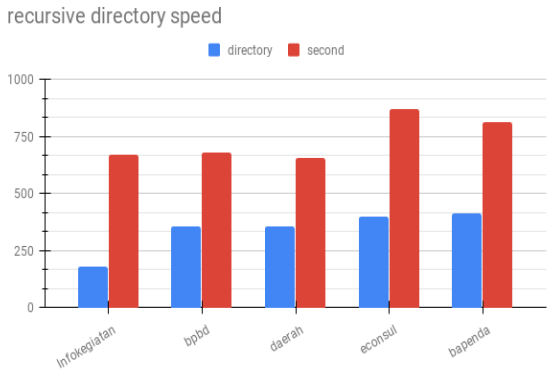
Gambar 6. Grafik Jumlah Files Dalam Subdirectory

4.2. Hasil Uji Coba Kecepatan Rekursif Pindai

Terlihat pada shelltrap melakukan scanning files membutuhkan waktu yang sebanding dengan jumlah total subdirectory Tabel 2, yang terdapat dalam masing-masing host. Hal ini terlihat pada kolom detik. Untuk visual atau grafik Kecepatan recursive subdirectory dapat dilihat pada Gambar 7.

Tabel 2. Kecepatan Rekursif Pindai

No	Host	Directory server	directory	files
1	Infokegiatan	/var/www/html/infokegiatan/public_html/	179	1085
2	bpbd	/var/www/html/bpbd/public_html/	355	2245
3	daerah	/var/www/html/daerah/public_html/	355	2234
4	econsul	/var/www/html/econsul/public_html/	399	2437
5	bapenda	/var/www/html/bapenda/public_html/	415	2664



Gambar 7. Visual Kecepatan Rekursif Directory

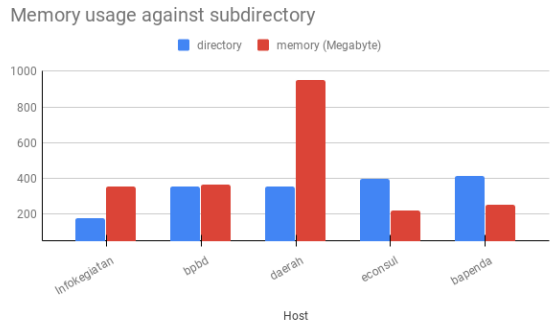
4.3. Hasil Penggunaan Memori saat Pindai

Shelltrap diujikan pula terhadap memori fisik yang ada pada server. Pengujian ini diperhatikan penting jika framework shelltrep tidak memerlukan memori yang cukup banyak

Tabel 3. Penggunaan Memori Terhadap Subdirectory

No	Host	Directory server	directory	files	Memory (kb)
1	Infokegiatan	/var/www/html/infokegiatan/public_html/	179	1085	355
2	bpbd	/var/www/html/bpbd/public_html/	355	2245	365
3	daerah	/var/www/html/daerah/public_html/	355	2234	952
4	econsul	/var/www/html/econsul/public_html/	399	2437	223
5	bapenda	/var/www/html/bapenda/public_html/	415	2664	256

Tabel 3, kecepatan deteksi backdor diukur oleh parameter memory yang digunakan untuk scan setiap subdirectory terhadap berapa jumlah waktu yang dibutuhkan. Untuk lebih jelas secara visual dapat dilihat pada Gambar 8.



Gambar 8. Grafik Penggunaan Memori Saat Pindai Subdirectory

5. Kesimpulan

Pada penelitian ini dijelaskan mengenai system deteksi backdoor named shelltrap. Shelltrap Tidak hanya mendeteksi backdoor shell PHP script tapi juga ASP script. Sebelum shelltrap mendeteksi secara sistematis mencirikan shell web melalui tiga aspek termasuk (i) interconnection attacker and server victim, (ii) realtime runing dilingkungan server, dan (iii) penggunaan sistem operasi sensitif. Dari hasil ujicoba didapatkan high detection rate 98 % dengan false positive rate 2 %.

Ucapan Terima Kasih

Terimakasih kepada Kementrian Riset dan Teknologi Indonesia, Kemenristekdikti. Telah mendukung Penelitian Dosen Pemula.

Daftar Pustaka

Bantun, S., Ashari, A., & Karim, R., 2020. Analisis Kinerja Raspberry Pi Sebagai SIP Server Untuk Aplikasi Video Phone. Semarang: Technocom.

Barth, A., Caballero, J., & Song, D., 2009. Secure content sniffing for web browsers, or how to stop papers from reviewing themselves. IEEE Symposium on Security and Privacy , 360-371.

Chao, W., Zhizhong, Wu, Z.W., Xi, L., Xuehai, Z., Aili, W. & Patrick C.K.H., 2014. SmartMal: a service-oriented behavioral malware detection framework for mobile devices. The Scientific World Journal , 2014 (101986,).

Duda, R., Hard, P., & D.G.S., 2012. Pattern Classification 2nd. In J. W. Sons. New York.

Eman, A., Amin, S., Mohammed, S., & Ayman, B.-E. 2017. A cloud-based malware detection framework. International Journal of Interactive Mobile Technologies , 11 (2), 113-127.

Garfinkel, S., 2007. Anti-Forensics: techniques, detection and countermeasures. 2nd International Conference in i-Warefare and Security , 79.

Haboob, T., 2018. File Upload Restrictions Bypass. Haboob.

Hung-Jen, L., Chun-Hung, R. L., Ying-Chih, L., & Kuang-Yuan, T., 2013. Intrusion detection system: A comprehensive review. Journal of Network and Computer Applications , 16-24.

Irfan, T.A. & Akbar, A.A.M., 2019. How To Using Online Meeting On Jitsi Meet Aplication. 1-13.

John, L.C., 2016. GVFS metadata: Shellbags for Linux. Digital Investigation , 16, 12 - 18.

Josh, A., Richard, B., Benton, C., Zakir, D., Peter, E., Alan, F.-L., 2019. Let's Encrypt: An Automated Certificate Authority to Encrypt the Entire Web. ACM SIGSAC Conference on Computer and Communications Security , 2019 (Session 10 E : Certificate), 2473 - 2487.

Khattab, M.A.A., Anna, G. and Klaus, D., 2015. An Intrusion Detection System Against Malicious Attacks on the Communication Network of Driverless Cars. Consumer Communications and Networking Conference (CNCC) , 916-921.

Kim, D.W., Yan, P., & Zhang, J., 2015. Detecting fake anti-virus software distribution webpages. Computers & Security , 49, 95 - 106.

Li, Y., Huang, J., Ikusan, A., Mitchell, M., Zhang, J., & Dai, R., 2019. ShellBreaker: automatically detecting PHP-based malicious web shells. Journal of Computers and Security , 87.

Lin, Y.-D., Chen, S.-H., Lin, P.-C., & Lai, Y.-C., 2008. Designing and evaluating interleaving decompressing and virus scanning in a stream-based mail proxy. The Journal of Systems and Software , 81, 1517 - 1524.

Maha A.S., 2020. Providing a secure environment for e-commerce sites using SSL Technology. Journal of Education and Science , 29 (1), 174 - 191.

Manesh, T., Brijith, B., Bhraguram, T. M., & Bhadran, V.K., 2013. Network forensic investigation of https protocol. International Journal of Modern Engineering Research (IJMER), 3 (5), 3096 - 3106.

McDaniel, P., & Rubin, A.D., 2005. Web security. Comput. Netw. United State: Elsevier.

Shweta, P., & Abhishek, S.C., 2013. Secure Content sniffing for web browsers. International Journal of Advanced Research in Computer and Communication Engineering , 2 (9), 3595-3601.

Simson, G., 2007. Anti-Forensics: techniques, detection and countermeasures, 2nd International Conference in i-Warefare and Security.

Son, S., & Shmatikov, V., 2011. Finding semantic vulnerabilities in PHP applications. Proceedings of the ACM SIGPLAN 6th Workshop on Programming Languages and Analysis for Security (pp. 1-13). NY, United States: Association for Computing Machinery.

Stiborek, J., Pevny, T., & Rehak, M., 2018. Probalistic Analysis of Dynamic Malware Traces. Computer & Security , 221-239.

Vazquez, A., 2016. Learn CentOS Linux Network Services. Apress, Berkeley, CA.

Waliulu, R. F., & Iskandar, A.T.H., 2018. Reverse engineering analysis forensic malware WEBC2-DIV. Jurnal &Penelitian Teknik Informatika) , 113 - 119.

Wassermann, G., & Su, Z., (2008. Static detection of cross-site scripting vulnerabilities. In: Proceedings of the 30th International Conference on Software Engineering , 171-180.

Xie, Y., & Aiken, A., 2016. Static Detection of Security Vulnerabilities in Scripting Languages. USENIX Security Symposium , 15, 179-192.

Xue, L., & Sun, G., 2014. Design and implementation of a malware detection system based on network behavior. Security and Communication Networks , 8, 459 - 470.

Yujie, F., Yanfang, Y., & Lifei, C., 2016. Malicious sequential pattern mining for automatic malware detection. Expert Systems With Applications , 16-25.

Zahra, S., Ashkan, S., & Mahboobe, G., 2017. MAAR: Robust features to detect malicious activity based on API calls, their arguments and return values. Engineering Applications of Artificial Intelligence , 93-102.