SECURITY HARDENING DENGAN CLOUD WEB SERVICE UNTUK PENGAMANAN WEBSITE BERBASIS WORDPRESS

Laurensius Faleddo Giri Retza[1], Affandy[2]
[1,2]sistem informasi, fakultas ilmu komputer, universitas dian nuswantoro
[1,2]Alamat, Semarang, 50131, telepon
E-mail : mail@faleddo.com[1], affandy@dsn.dinus.ac.id[2]

Abstrak

Perkembangan jumlah website dari tahun ke tahun selalu meningkat. Salah satu CMS yang paling banyak digunakan untuk mengembangkan website adalah Wordpress. Banyaknya pengguna Wordpress membuat hacker tertarik untuk mengeksploitasi celah keamanan pada Wordpress. Jumlah website berbasis Wordpress yang diretas setiap tahunnya selalu meningkat. Hal ini sangat merepotkan para pengelola website, terutama bagi mereka yang mengelola lebih dari satu website yang harus memperbaiki websitenya satu persatu. Untuk mengatasi hal tersebut, dibuatlah suatu metode keamanan yang memungkinkan para pengelola website tersebut untuk lebih mudah melakukan keempat tahapan security hardening yaitu Access, Analyze, Remediate, dan Manage guna mengamankan websitenya dari satu tempat terintegrasi dengan memanfaatkan aplikasi keamanan berbasis cloud. Berdasarkan survei pengguna, aplikasi ini berhasil mempermudah dan mempersingkat waktu pengguna dalam melakukan pengamanan pada website.

Kata Kunci: *website, keamanan, Wordpress, cloud, security hardening*

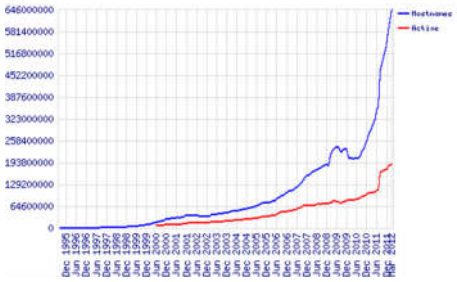
Abstract

Websites growth from year by year is always increasing. One of the most widely used CMS to develop a website is Wordpress. The big number of WordPress users make hackers interested to exploit security holes in Wordpress. Hacked WordPress-based websites is always increasing every year. It is very annoying website's owners, especially for those who manage more than one website which should fix their website one by one. To solve this problem, this research propose a security method that allows website managers to easily perform four security hardening stages: Access, Analyze, Remediate, and Manage to secure the website within one integrated cloud-based security applications. Based on user acceptance test, the application is successfully simplify and shorten the time to harden security on Wordpress based websites.

Keywords: *websites, security, Wordpress, cloud, security hardening*

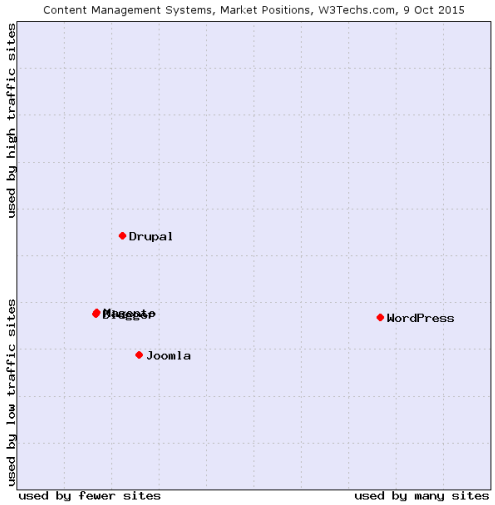
1. PENDAHULUAN

Perkembangan jumlah situs web pada era teknologi informasi saat ini menunjukkan tren peningkatan yang begitu cepat. Menurut survei yang diadakan oleh Netcraft, sebuah perusahaan riset dan keamanan internet, jumlah situs web yang aktif saat ini mencapai 644 juta situs web. Lebih tepatnya 644,275,754 situs web [1]. Dan angka tersebut akan terus bertambah setiap tahunnya. Semakin mudahnya orang memiliki situs web mendukung meningkatnya jumlah situs web saat ini. Salah satunya adalah penggunaan Content Management System (CMS) yang memungkinkan pengguna membuat situs web hanya dalam beberapa menit saja.



Gambar 1 Statistik peningkatan jumlah situs web

Wordpress, merupakan sebuah *Content Management System* (CMS) yang banyak digunakan untuk mengembangkan situs web. Mulai situs web pribadi, blog, toko online, hingga situs web korporasi dan pemerintahan. Berdasarkan survei dari W3techs, sebuah perusahaan survei teknologi web, *marketshare* Wordpress saat ini mencapai 24,5% dari total situs web yang tersedia di internet. Sedangkan dari situs web yang menggunakan CMS lain seperti Joomla, Drupal, dan sebagainya, Wordpress menempati 58,9% posisi situs web dengan teknologi CMS [2].



Gambar 2 Statistik marketshare CMS

Dengan semakin meningkatnya minat pemilik situs web termasuk perusahaan komersial dan pemerintahan terhadap Wordpress karena kemudahan penggunaannya dan sifatnya yang open-source, semakin banyak pula pengembang web yang menciptakan plugin untuk menambah fungsionalitas CMS Wordpress dan berbagai macam tema untuk kustomisasi tampilan situs web berbasis Wordpress. Dampaknya, hal itu semakin membuka banyak celah keamanan terhadap *core* sistem Wordpress itu sendiri. Seperti dari plugin atau tema yang memiliki celah keamanan, atau plugin atau tema premium bajakan yang disusupi *backdoor*.

Menurut survei The Hacker News, sebuah portal keamanan informasi, pada tahun 2009, jumlah situs web berbasis Wordpress yang terkena serangan hacker mencapai 81.00 situs web. Angka ini terus meningkat hingga pada tahun 2012 jumlahnya mencapai 170.000 situs web [3]. Dibutuhkan setidaknya waktu lima hari untuk mencari kerusakan memperbaiki seluruh kerusakan pada satu situs web dan biaya sekitar 100 US dollar untuk jasa perbaikan [4]. Apabila dalam setahun ada 100.000 situs web yang diretas, maka diperkirakan akan

membutuhkan waktu yang banyak untuk membenahi situs web tersebut dan biaya 100.000.000 US Dollar lebih untuk biaya jasa maintenance. Itupun belum termasuk kerugian akibat berhentinya proses bisnis perusahaan saat sistemnya diretas. Berdasarkan kondisi yang terjadi saat ini diketahui bahwa WordPress rentan diserang oleh hacker karena plugin dan temanya yang terdapat celah keamanan. Ulah hacker yang merusak situs-situs web ini mengakibatkan kerugian materi dan non-materi yaitu kerugian berupa besarnya biaya yang harus dikeluarkan untuk jasa maintenance website dan kerugian berupa waktu yang terbuang hanya untuk membenahi situs web yang diretas, apalagi perusahaan yang menangani lebih dari satu situs web seperti provider web hosting. Dari paparan di atas, permasalahan pada penelitian ini adalah bagaimana cara mengamankan situs web berbasis WordPress dengan cepat dan efektif, terutama bagi para pemilik situs web yang mengelola lebih dari satu situs web. Untuk itu, diperlukan sebuah pendekatan untuk mengamankan situs web yang berbasis Wordpress dari serangan peretas dengan cara yang lebih cepat dan efisien, melalui pengembangan satu portal aplikasi untuk memantau dan melakukan maintenance pada lebih dari satu situs web sekaligus.

2. LANDASAN TEORI

2.1 Content Management System

Diawali pada tahun 1997 oleh Kasper Skårhøj yang mengembangkan Content Management System (CMS) bernama TYPO3, yang pada tahun 1999 mulai disebarluaskan dengan lisensi open source. CMS sendiri merupakan sebuah sistem pada aplikasi web yang digunakan untuk memudahkan pengguna mengelola isi sebuah website secara dinamis melalui sebuah tampilan yang user-friendly tanpa harus mengedit kode web .

2.2 Proteksi Aset Informasi

Bagi perusahaan atau instansi, informasi merupakan salah satu aset yang sangat penting untuk dilindungi dari pihak-pihak yang tidak bertanggung-jawab karena informasi-informasi tersebut dapat berpengaruh bagi suatu instansi mulai dari perencanaan hingga proses bisnis instansi tersebut.

Proteksi aset informasi adalah salah satu usaha untuk melindungi dan menentukan solusi masalah keamanan aset informasi, khususnya dari sisi teknologi [12]. Salah satu teknik pengamanan untuk melakukan proteksi aset informasi adalah security hardening.

2.3 Cloud Computing

Cloud Computing merupakan ide dari seorang pakar Komputasi dan Intelegensi Buatan Massachusetts Institute of Technology (MIT), John McCarthy pada tahun 1960 yang menyatakan bahwa nantinya komputer dan internet akan menjadi insfrastuktur publik seperti telepon dan listrik. Konsepnya, kita tidak perlu memiliki perangkat lunak pada perangkat kita tapi kita cukup mengakses server untuk menggunakan perangkat lunak tersebut yang mendasari ide terbentuknya SalesForce, layanan CRM berbasis *cloud computing* pada tahun 2000. Kemudian sejak tahun 2005, *cloud computing* terus berkembang tidak hanya pada lingkup aplikasi ataupun sistem operasi, tetapi juga infrastruktur fisik [15].

2.4 Application Programming Interface (API)

API adalah sebuah fungsi di mana dua atau lebih aplikasi yang berbeda bertukar data untuk mendukung fungsinya satu sama lain [16]. Dengan adanya API, selain untuk pertukaran data, pengembang aplikasi pihak ketiga juga

bisa membuat aplikasi baru dengan layanan API yang disediakan oleh penyedia aplikasi.

2.6 Web

Teknologi web bermula pada tahun 1989 yang ditemukan oleh Tim Berners dan beberapa peneliti lainnya mengenai protokol pertukaran informasi melalui jaringan internet. Saat ini teknologi web memungkinkan jutaan pengguna di dunia saling terkoneksi dan dapat berkomunikasi melalui aplikasi-aplikasi berbasis web dan berbagai macam jenis layanannya [18].

3. SECURITY HARDENING

Berikut merupakan keempat tahapan dalam security hardening:

3.1 Access

Mengidentifikasi celah-celah keamanan yang masih terdapat dalam sistem dan mengeksploitasi sistem untuk mengetahui lubang keamanan pada sistem tersebut.

3.2 Analyze

Memperkirakan tingkat keamanan dan menganalisis dampak yang terjadi pada celah keamanan tersebut, kemudian mengklasifikasikan tingkat kerusakan yang diakibatkan oleh celah keamanan tersebut.

3.3 Remediate

Menemukan celah keamanan pada sistem yang diuji dan mencari cara untuk menutup celah keamanan tersebut untuk mengamankan sistem.

3.4 Manage

Mengintegrasikan patch pada celah keamanan untuk menutup lubang keamanan dan mencegah serangan masuk, serta mencegah terbukanya celah keamanan lain.

4. HASIL DAN PEMBAHASAN

Security hardening terdiri dari 4 tahapan. Antara lain tahap *Access* untuk mencari celah keamanan pada sistem atau mengeksploitasi sistem. Yang kedua, tahap *Analyze* untuk menganalisis dampak dari kelemahan tersebut dan mengklasifikasi tingkat bahayanya. Tahap selanjutnya, mencari solusi untuk menutup celah keamanan tersebut dan tahap terakhir yaitu mengelola sistem tersebut dan mencegahnya agar tidak terjadi peretasan dan ditemukan celah keamanan lagi. Berikut ini penjabaran dari keempat tahapan *security hardening* pada sistem CMS Wordpress.

4.1 Access

Tahap pertama dalam *security hardening* adalah mencari celah keamanan pada sebuah sistem. Pada CMS Wordpress, celah yang biasa ada pada CMS menurut Wordpress Security WhitePaper antara lain:

4.1.1 SQL Injection

SQL injection adalah salah satu cara meretas website dengan memanfaatkan kelalaian developer dalam mem-filter *query database* pada aplikasi yang dibuatnya sehingga *hacker* dapat dengan mudah meng-*inject query* SQL melalui parameter GET di website.

Pada Wordpress sendiri, SQL *injection* disebabkan oleh *plugin* atau *themes* dari pihak ketiga yang diinstall pada CMS ini. Sedangkan celah SQL *injection* yang disebabkan oleh *core* Wordpress sendiri sangat jarang terjadi.

4.1.2 Broken Authentication and Session Management

Wordpress menggunakan Cookies pada PHP untuk mengelola sesi dan hak akses pengguna. Sedangkan data *username* dan *password* disimpan di *database* dengan enkripsi PHPass. Meskipun sudah menggunakan teknologi enkripsi yang baik, apabila pengguna lengah dalam mengatur autentikasi pengguna

maka Wordpress juga dapat menjadi rentan terkena serangan

4.1.3 Cross Site Scripting (XSS)

Cross Site Scripting pada Wordpress biasa ditemukan pada *form* input di Wordpress. Tidak hanya pada *core* Wordpress saja, tetapi juga pada *plugin* Wordpress yang memiliki fitur *input form* dan *theme* Wordpress yang memiliki kelemahan dalam menangani *user interface*.

4.1.4 Insecure Direct Object Reference

Celah ini terjadi karena halaman administrator Wordpress yang dapat diakses oleh siapa saja pada pengaturan *default* padahal halaman ini seharusnya hanya dapat diakses oleh orang tertentu saja. Apalagi jika *link default* pada Wordpress masih sama seperti kondisi *default*nya yaitu '/wp-admin'. Saat hacker mengakses /wp-admin, maka halaman *login* akan langsung muncul. Hal ini memungkinkan seorang *hacker* untuk melakukan *brute-force attack* pada Wordpress.

4.1.5 Security Misconfiguration

Kesalahan konfigurasi paling sering terjadi pada Wordpress. Hal ini biasanya diakibatkan karena kelalaian pengguna setelah melakukan instalasi atau *maintenance* tidak mengembalikan pengaturan semula atau tidak mengatur keamanannya lebih lanjut.

Misal, pengaturan *mode development* dan *mode production*. Jika pengguna lalai masih mengaktifkan *mode development*, maka jika terjadi *error* Wordpress akan menampilkan semua *debug message* yang mengakibatkan *hacker* mengetahui dimana celah keamanan pada website itu

4.1.6 Sensitive Data Exposure

Celah ini disebabkan karena kelalaian *developer* yang tidak teliti mengamankan data-data yang seharusnya tidak diketahui pengunjung seperti respon dari AJAX yang tiba-tiba muncul di tampilan depan, password konfigurasi yang dicatat di komentar bahasa pemrograman yang lupa dihapus sehingga pengunjung dapat menemukan akses *login* dengan melihat *source code* pada *browser*. Informasi versi *software* yang digunakan juga dapat menjadi bahan analisis *hacker* sebelum menyerang Wordpress

4.1.7 Missing Function Level Access Control

Wordpress memiliki fitur dimana pengguna dapat memiliki hak akses yang berbeda-beda dalam satu *website*. Misal pengguna A dapat menulis *posting* dan mengelola komentar sedangkan pengguna B hanya dapat menulis *posting* saja. Dengan kelemahan ini, apabila administrator lupa membatasi hak akses, maka pengguna B juga dapat mengelola komentar tanpa perlu mengetahui *password* pengguna A. Atau apabila administrator keliru memberikan hak akses administrator ke semua akun pengguna, apabila pengguna *login* dia dapat mengakses semua fitur administrator. Bahkan dia juga bisa menghapus akun administrator yang asli.

4.1.8 Cross Site Request Forgery (CSRF)

Celah CSRF disebabkan karena *form input* yang tidak diberi *token* untuk mengamankan hasil inputan. Dengan celah ini, penyerang dapat mengirimkan banyak data ke *form website* tanpa harus memiliki hak akses pengguna atau tanpa melalui *form input* yang sebenarnya pada *website*. Yang lebih berbahaya, penyerang juga dapat melakukan *privilege escalation* atau mengubah hak akses user biasa menjadi administrator.

4.1.9 Using Component With Known Vulnerability

Tidak hanya plugin dan theme yang memiliki celah keamanan, tetapi juga

komponen pihak ketiga yang digunakan Wordpress atau *plugin*nya. Komponen yang dimaksud seperti komponen What You See is What You Get (WYSIWYG) pada editor teks Wordpress.

4.1.10 Unvalidated Redirects and Forwards

Wordpress menggunakan *redirect* untuk mengarahkan penggunanya ke halaman tertentu, terutama pada halaman administrator. Kesalahan pengaturan *redirect* dapat mengakibatkan pengguna biasa tidak sengaja di-*redirect* ke halaman yang tidak seharusnya pengguna akses. Akibatnya, pengguna bisa diarahkan ke situs yang tidak seharusnya dibuka misalkan situs penipuan atau situs yang menyebarkan *malware*

4.2 Analyze

Tahap kedua dari *security hardening* adalah menganalisa dampak dari celah-celah keamanan tersebut, mengklasifikasi tingkat bahayanya, kemudian memperkirakan dampak apa saja yang terjadi jika celah keamanan tersebut berhasil dieksploitasi oleh penyerang. Klasifikasi tingkat bahaya keamanan ini berdasarkan pada:

1. ***Loss of confidentiality***, apakah data yang terancam memiliki tingkat kerahasiaan yang tinggi.
2. ***Loss of integrity***, apakah serangan dapat mengakibatkan kerusakan pada data.
3. ***Loss of availability***, apakah serangan dapat mengakibatkan layanan terinterupsi/server *down*.
4. ***Loss of accountability***, apakah pelaku serangan dapat dengan mudah dilacak.

Masing masing aspek tersebut memiliki satu poin penilaian yang akan dijumlahkan untuk mengukur tingkat ancaman keamanan. Dari analisis kesepuluh celah keamanan diatas, bisa di simpulkan dalam tabel sebagai berikut:

Tabel 4.1 Klasifikasi bahaya celah keamanan pada Wordpress

No.	Celah Keamanan	Poin	Klasifikasi
1	SQL Injection	4	Kritis
2	Broken Authentication and Session Management	3	Tinggi
3	Cross Site Scripting (XSS)	3	Tinggi
4	Insecure Direct Object Reference	3	Tinggi
5	Security Misconfiguration	4	Kritis
6	Sensitive Data Exposure	4	Kritis
7	Missing Function Level Access Control	3	Tinggi
8	Cross Site Request Forgery (CSRF)	4	Kritis
9	Using Component With Known Vulnerability	3	Tinggi
10	Unvalidated Redirects and Forwards	3	Tinggi

Dari tabel di atas, bisa disimpulkan bahwa SQL *Injection*, *Security Misconfiguration, Sensitive Data Exposure*, dan *Cross Site Request Forgery* (CSRF) memiliki tingkat kerusakan yang kritis. Hal ini disebabkan karena keempat celah tersebut selain memiliki dampak kerusakan yang tinggi, celah ini juga

berpotensi membuka celah keamanan yang lain bagi *hacker*. Misalkan dari celah *Sensitive Data Exposure*. Karena celah ini, *hacker* bisa menganalisis kelemahan Wordpress yang lain yang mengakibatkan dampak kerusakan yang lebih besar seperti XSS dan CSRF sekaligus.

4.3 Remediate

Tahapan selanjutnya dari *security hardening* adalah mencari solusi dan metode untuk menutup celah keamanan yang ada atau memperbaiki konfigurasi sistem yang kurang optimal untuk keamanan.

4.4 Manage

Tindakan terakhir dari *security hardening* adalah mengelola sistem agar tetap aman dan memantaunya agar tidak terkena serangan lagi. Berikut ini adalah langkah-langkah memantau dan mengelola keamanan pada CMS Wordpress.

Mayoritas celah keamanan pada sistem CMS Wordpress diakibatkan oleh *plugin* dan *theme* dari pihak ketiga terutama *plugin* dan *theme open source* yang tersedia gratis di *repository* Wordpress. Untuk menjaga keamanan *plugin* dan *theme* ini, sebaiknya pengguna tidak menginstall *plugin* dan *theme* sembarangan sebelum melihat *review* dari pengguna lain dan *rating* dari *plugin* atau *theme* tersebut di *repository* Wordpress.

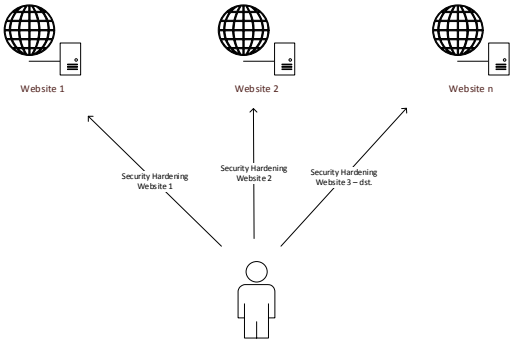
Untuk celah keamanan dari *core* CMS Wordpress sendiri bisa dicegah dengan selalu melakukan *update* versi Wordpress ke versi *stable* terbaru. Penggunaan *update* versi *nightly build* biasanya kurang sempurna karena belum dirilis secara resmi untuk mode *production*, sekedar versi *preview* untuk para *developer plugin* dan *theme*. Untuk penggunaan sehari-hari, sebaiknya pengguna biasa menggunakan versi *stable*.

Sedangkan untuk celah yang diakibatkan kelalaian pengguna karena kesalahan konfigurasi atau penggunaan konfigurasi yang kurang optimal dicegah dengan melakukan pemantauan secara rutin. Pemantauan juga bisa dilakukan dengan bantuan *plugin* pihak ketiga seperti Bulletproof Wordpress Security, Acunetix Wordpress Security, WP-Secure, Login Lockdown, dan sebagainya. Akan tetapi, pemantauan menggunakan *plugin* tersebut tetap merepotkan apabila pengguna memiliki lebih dari satu *website*. Untuk mengatasi hal itu, peneliti menyarankan sebuah model keamanan sebagai berikut ini yang dapat mengefisiensi waktu untuk menjaga keamanan dari CMS Wordpress.

4.5 Pengembangan Model Keamanan

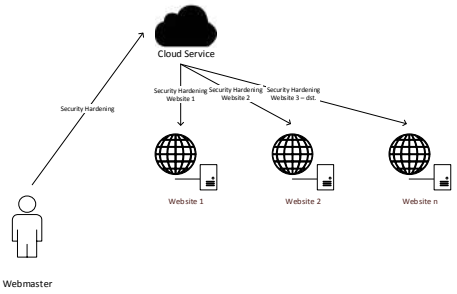
Berdasarkan uraian di atas, dari sepuluh celah keamanan utama menurut OWASP, empat diantaranya masuk dalam kategori kritis karena dampaknya yang mencakup keempat faktor penilaian yaitu *Loss of confidentiality, Loss of integrity, Loss of availability*, dan *Loss of accountability*. [11]

Aktivitas *recovery* atau perbaikan dari kerusakan yang diakibatkan celah tersebut cukup banyak. Hal ini menurut penulis cukup *merepotkan* administrator *website*, apalagi jika administrator mengelola lebih dari satu *website* seperti seorang *webmaster*, *provider hosting*, dan lain sebagainya. Hal ini bisa diilustrasikan sebagai berikut.



Gambar 4.3 Aktivitas webmaster melakukan security hardening

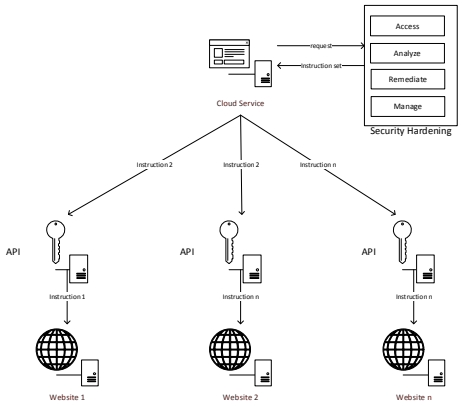
Untuk mengatasi hal tersebut, penulis mengusulkan sebuah model keamanan dengan ilustrasi sebagai berikut.



Gambar 4.4 Ilustrasi model security hardening

Dengan adanya konsep *security hardening* tersebut, *webmaster* tidak perlu lagi melakukan *hardening* satu persatu melainkan cukup melakukan *hardening* melalui sebuah *cloud service* lalu *cloud service* tersebut yang akan melakukan *hardening* ke masing-masing *website* yang dikelola oleh *webmaster* tersebut.

Adapun secara teknis, proses *security hardening* yang dilakukan oleh *cloud service* tersebut diilustrasikan sebagai berikut.



Gambar 4.5 Rancangan teknis security hardening berbasis SaaS Cloud

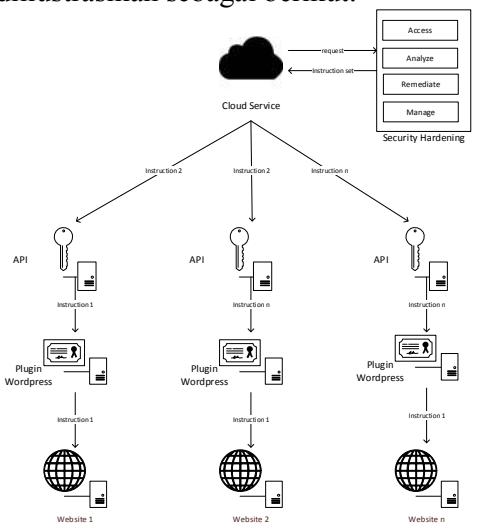
Dengan model keamanan ini, *webmaster* cukup mengirimkan spesifikasi kebutuhan keamanannya ke layanan SaaS *cloud* kemudian layanan ini akan melakukan *security hardening* berdasarkan spesifikasi webmaster. Kumpulan dari instruksi untuk melakukan *hardening* tadi, akan dikirimkan oleh layanan *cloud* ke masing-masing *website* yang dikelola oleh *webmaster* melalui perintah yang diterjemahkan dengan API untuk menghubungkan *website* dengan *cloud service*.

Pada CMS Wordpress, pemanfaatan API dilakukan dengan sebuah *plugin* untuk menghubungkan *core* Wordpress dan fungsi-fungsi pada Wordpress dengan instruksi dari *cloud service*. Untuk mempermudah penggunaan *cloud service* ini, penulis juga membuat prototype layanan SaaS *cloud* berbasis web.

4.6 Prototype Aplikasi Keamanan

Untuk membantu proses *security hardening* pada banyak *website* sekaligus, penulis membuat *prototype* aplikasi berbasis SaaS *cloud* yang berfungsi untuk mengotomasi *security hardening* pada *website-website* berbasis

Wordpress. Arsitektur aplikasi ini dapat diilustrasikan sebagai berikut:



Gambar 4.6 Arsitektur aplikasi Cloud

Aplikasi ini terdiri dari 3 bagian yaitu bagian utama layanan *Cloud* yang berfungsi sebagai *server* pusat untuk mengendalikan keamanan *website-website* dan untuk menyediakan *interface* bagi pengguna untuk mengelola *website-website*nya. Bagian kedua yaitu API untuk menerjemahkan perintah-perintah *security hardening* dari *server cloud* ke perintah yang dapat dikenali oleh Wordpress. Sedangkan bagian terakhir adalah sebuah *plugin* Wordpress yang berfungsi untuk mengenali perintah-perintah yang dikirimkan oleh API dari *cloud server* ke Wordpress dan mengimplementasikan *security hardening* pada *website* berbasis Wordpress yang terpasang *plugin* tersebut.

4. KESIMPULAN

Berdasarkan penelitian dan pembahasan yang telah dilakukan pada bab sebelumnya, maka dapat diambil kesimpulan bahwa penggunaan aplikasi berbasis cloud untuk memantau dan mengelola lebih dari satu website dapat mempermudah pekerjaan *webmaster*, *web developer*, maupun administrator yang mengelola lebih dari satu *website* untuk memantau dan mengamankan *website-website*nya. Selain mempermudah, aplikasi ini juga mengefisiensi pekerjaan mereka karena waktu yang digunakan untuk mengelola lebih dari satu *website* kini bisa mereka lakukan lebih cepat.

Akan tetapi, terdapat kendala pada tahap *remediate* yaitu pada saat aplikasi mencoba mengubah konfigurasi pada web *server* yang kurang optimal untuk keamanan karena *plugin* Wordpress hanya memiliki hak akses sebatas pada *core* Wordpress saja sehingga hanya celah pada CMS Wordpress saja yang diamankan, tidak sampai pada keamanan web *server*.

Selain itu, *plugin* tidak dapat melakukan *update core* Wordpress dan *plugin*/*theme* secara otomatis tetapi hanya dapat memberitahukan pada pengguna jika versi Wordpressnya sudah lama dan ada *update* terbaru dikarenakan untuk melakukan *update script* Wordpress harus melalui akses FTP atau melalui halaman *back-end* Wordpress sendiri, tidak bisa melalui *plugin* pihak ketiga Wordpress.

Akan tetapi, aplikasi keamanan berbasis *cloud* ini masih memiliki banyak kekurangan seperti kompatibilitasnya yang hanya terbatas pada CMS Wordpress dan server berbasis Apache. Adapun saran dari penulis antara lain:

1. Kedepannya fitur Wordpress yang akan datang tentunya membawa perubahan dan tambahan API baru yang mungkin mempermudah akses pengembang *plugin* pihak ketiga untuk lebih mudah mengontrol keamanan Wordpress melalui aplikasi di luar *core* Wordpres sehingga dapat dibuat pengamanan yang lebih ketat dengan memanfaatkan API terbaru Wordpress.

2. Meningkatkan kemampuan *tunnel*/*plugin* untuk mengakses FTP server untuk melakukan tindakan pengamanan lebih lanjut menggunakan *client* FTP berbasis PHP karena dengan *tool* FTP, hak akses modifikasi *file* ke server menjadi lebih bebas dibanding hanya dengan memanggil fungsi-fungsi *native* bawaan PHP.

DAFTAR PUSTAKA

[1] Business Insider, "How Many Web Sites Are Are There?," March 2012. [Online]. Available: www.businessinsider.com/how-many-web-sites-are-are-there-2012-3.

[2] W3techsUsage statistics and market share of WordPress for websitesW3techs24September2015Available:http://w3techs.com/technologies/details/cm-wordpress/all/all24September2015

[3] StartBloggingOnline, "What If Your WordPress Gets Hacked," New York, 2013.

[4] The Hacker News, "45000 Wordpress blogs hacked on 2nd day of Spam campaign:," The Hacker News, October 2012. [Online]. Available: http://thehackernews.com/2012/10/45000-wordpress-blogs-hacked-on-2nd-day.html. [Diakses 26 September 2015].

[5] R. Rahmadi, *Studi Komparatif Penggunaan Open Source Content Management System (CMS) Joomla Dan Drupal Untuk Pembuatan Website,* 2010.