PANDUAN KEAMANAN
HARDENING WORDPRESS

DIREKTORAT OPERASI KEAMANAN SIBER
NATIONAL CSIRT OF INDONESIA
Id-SIRTI/CC
INDONESIA SECURITY INCIDENT RESPONSE TEAM ON INTERNET INFRASTRUCTURE COORDINATION CENTER

Ditandatangani secara elektronik oleh:
Koordinator Kelompok/Fungsi Operasi Deteksi, Penanggulangan dan Pemulihan, Penanganan Insiden dan Krisis Siber Nasional

Taufik Arianto, S.ST., M.Kom.
Pembina (IV/a)

versi: 1.0
TLP: WHITE

DAFTAR ISI

01 GAMBARAN UMUM

02 APA ITU KEAMANAN?

03 TEMA KEAMANAN

04 KERENTANAN KOMPUTER

04 KERENTANAN WORDPRESS

06 KERENTANAN WEB SERVER

06 KERENTANAN JARINGAN

08 PASSWORD

10 FTP

11 PERIZINAN FILE

15 KEAMANAN DATABASE

18 PENGAMANAN WP-ADMIN

21 PENGAMANAN WP-INCLUDE

22 PENGAMANAN WP-CONFIG.PHP

23 PENGAMANAN PENGEDITAN FILE

24 PLUGIN

26 SECURITY THROUGH OBSCUTITY

27 CADANGAN DATA

28 LOGGING

30 MONITORING

Hardening Wordpress

GAMBARAN UMUM

WordPress memerlukan suatu keamanan seperti halnya sistem lain, karena pasti selalu ada potensi kerentanan yang muncul jika tindakan-tindakan pencegahan tidak dilakukan. Dokumen ini merupakan panduan yang membahas beberapa bentuk kerentanan umum dan hal-hal yang dapat dilakukan untuk membantu menjaga keamanan instalasi WordPress.

WordPress

APA ITU KEAMANAN?

Apa yang dimaksud dengan keamanan adalah pengurangan risiko, bukan penghapusan risiko. Hal ini tentang menerapkan semua kontrol yang sesuai dan tersedia dengan alasan untuk mengurangi kemungkinan menjadi target peretasan.

WEBSITE HOST

Hosting menjadi hal pertama untuk diamankan pada *website*. Panduan ini menjelaskan antara *website host* dan keamanan *website.* Server yang aman tentu saja dapat melindungi data pribadi, integritas, dan ketersediaan sumber daya di server administrator.

Berikut ini bentuk *website host* yang aman:

- Kemudahan diskusi dengan *website host* terkait keamanan dan fitur proses keamanan *hosting* yang ditawarkan.
- Tersedianya versi terbaru pada semua perangkat lunak (*software*).
- Tersedianya *back up* (cadangan) dan *recovery* (pemulihan).

Website Host bertanggung jawab pada infrastruktur *website*, namun tidak bertanggung jawab pada aplikasi yang terpasang.

KONSEP KEAMANAN

Berikut ini beberapa aspek keamanan yang dapat dijadikan pertimbangan:

PEMBATASAN AKSES

Pembatasan akses merupakan upaya untuk mengurangi kemungkinan pihak-pihak yang tidak berkepentingan mengakses sistem.

CONTAINMENT

Containment yaitu pengkonfigurasian sistem untuk meminimalkan jumlah kerusakan ketika terjadi peretasan.

PERSIAPAN DAN PENGETAHUAN

Selalu menyimpan *back up* data dan mengetahui status instalasi WordPress dengan memeriksanya secara berkala. Memiliki perencananan *back up* dan memulihkan instalasi apabila terjadi bencana sehingga dapat membantu *online* lebih cepat.

SUMBER TERPERCAYA

Penggunaan *plugin*/tema disarankan menggunakannya dari sumber terpercaya dan menggunakan situs WordPress.org atau perusahaan resmi yang sudah terkenal.

KERENTANAN KOMPUTER

Perlu untuk memastikan komputer yang digunakan terbebas dari *spyware*, *malware*, dan virus. Tidak ada jumlah keamanan di WordPress atau di server web yang akan membuat perbedaan sedikit pun jika ada *keylogger* di komputer.

Selalu lakukan pembaruan sistem operasi dan perangkat lunak yang terinstal pada komputer, terutama *browser web*. Jika menjelajahi situs yang tidak tepercaya, sebaiknya menggunakan alat seperti tanpa *script* (atau nonaktifkan javascript/flash/java) di browser.

KERENTANAN WORDPRESS

WordPress diperbarui secara berkala untuk mengatasi masalah keamanan yang mungkin muncul. Meningkatkan keamanan perangkat lunak menjadi perhatian yang berkelanjutan, Selalu mengikuti perkembangan WordPress versi terbaru.

SUMBER TERPERCAYA

Versi terbaru WordPress selalu tersedia dari situs utama WordPress di https://wordpress.org. Perilisan versi terbaru yang resmi tidak akan tersedia di situs lain, untuk itu sangat disarankan tidak mengunduh atau menginstal WordPress dari *website* apa pun selain https://wordpress.org.

Sejak versi 3.7, WordPress telah memberikan fitur terbaru berupa pembaruan otomatis. Fitur ini disarankan untuk digunakan karena memudahkan proses pembaruan. Dashboard WordPress juga perlu untuk dimanfaatkan agar terus mendapatkan informasi tentang pembaruan.

Apabila ditemukan kerentanan pada WordPress, informasi yang dapat digunakan peretas untuk mengeksploitasi kerentanan pasti sudah tersebar. Hal tersebut membuat versi lama WordPress lebih rentan untuk diserang. Dengan begitu pemilik *website* perlu melakukan pembaruan sesuai dengan perilisan terbaru dari WordPress.

Jika pemilik *website* merupakan administrator yang bertanggung jawab atas lebih dari satu instalasi WordPress, perlu untuk mempertimbangkan penggunaan Subversion yang akan mempermudah pemilik *website* untuk mengelola.

PELAPORAN ISU KEAMANAN

Pelaporan isu keamanan dilakukan jika menemukan kelemahan keamanan pada WordPress, maka diharapkan untuk melaporkan masalah tersebut kepada pihak WordPress. Tata cara pelaporan terdapat pada FAQ situs resmi WordPress.

Hal tersebut juga berlaku jika menemukan *bug* pada aplikasi WordPress. Petunjuk pelaporan *bug* juga tersedia di situs resmi Wordpress.

KERENTANAN WEB SERVER

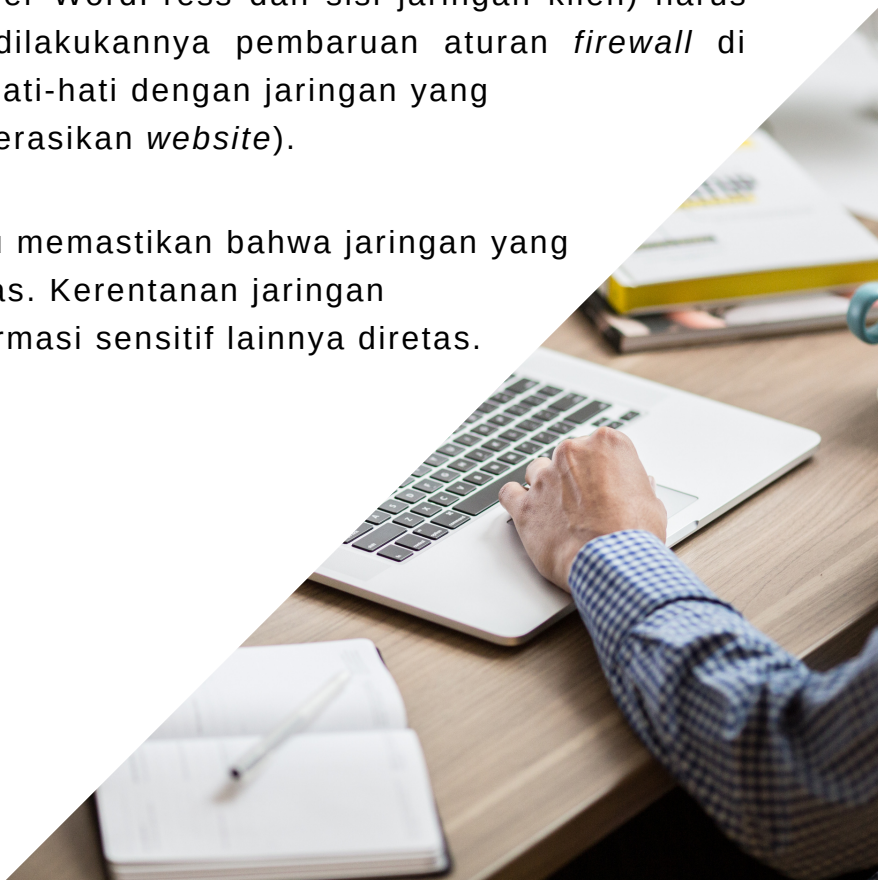
Web server yang menjalankan WordPress dan perangkat lunak di dalamnya dapat memiliki kerentanan. Oleh karena itu, perlu untuk memastikan versi server web dan perangkat lunak yang dijalankan aman dan stabil, atau pastikan menggunakan *host* tepercaya yang menangani hal-hal ini.

Apabila *website* yang berada di *shared server* (server bersama yang digunakan untuk meng-*hosting website* lain selain milik pribadi) diretas, maka *website* pribadi juga berpotensi untuk diretas meskipun sudah mengikuti langkah-langkah pada panduan ini. Maka, perlu memastikan dengan bertanya kepada *website host* untuk tindakan pencegahan keamanan.

KERENTANAN JARINGAN

Jaringan di kedua ujung (sisi server WordPress dan sisi jaringan klien) harus dapat dipercaya. Artinya, perlu dilakukannya pembaruan aturan *firewall* di router milik pribadi dan selalu berhati-hati dengan jaringan yang terkoneksi ketika bekerja (mengoperasikan *website*).

Host web dan pemilik *website* perlu memastikan bahwa jaringan yang diguakan tidak disusupi oleh peretas. Kerentanan jaringan memungkinkan kata sandi dan informasi sensitif lainnya diretas.



BAGAIMANA CARA *HARDENING WORDPRESS?*

PASSWORD

Password atau kata sandi adalah aspek penting pada *hardening*, tujuan penggunaan kata sandi adalah mempersulit orang lain untuk menebak dan mempersulit serangan dari peretas seperti *brute-force attack.* Banyak alat pembuat kata sandi otomatis untuk membuat kata sandi yang aman.

WordPress juga dilengkapi dengan fitur pengukur kekuatan kata sandi yang ditampilkan saat mengubah kata sandi di WordPress. Fitur ini disarankan untuk digunakan saat mengubah kata sandi untuk memastikan kekuatan kata sandi yang memadai.

Hal-hal yang harus dihindari saat memilih kata sandi:

- Penggunaan kombinasi atau permutasi dari nama asli, nama pengguna, nama perusahaan, atau nama *website*.
- Kata-kata dari kamus, dalam bahasa apapun.
- Kata sandi singkat.
- Kata sandi hanya numerik atau abjad saja (campuran keduanya adalah yang kombinasi kata sandi yang baik).

Kata sandi kuat diperlukan tidak hanya untuk melindungi konten blog namun untuk menghindari peretas mendapatkan akses akun administrator. Hal ini perlu dihindari karena jika peretas mendapatkan akses ke akun administrator, peretas dapat memasang *script* berbahaya dan berpotensi membahayakan seluruh server.

Selain menggunakan kata sandi yang kuat, sebaiknya gunakan *two-factor authentication* sebagai tindakan keamanan tambahan.

Two-factor authentication berarti menambahkan persyaratan kedua. Persyaratan kedua yang dimaksud yaitu adanya kode yang dikirimkan ke perangkat (*handphone*, tablet, dll), sehingga seseorang tidak dapat masuk ke situs web apabila tidak memegang perangkat tersebut.

Berikut ini langkah-langkah dalam mengimplementasikan *two-factor authentication*

- Cari 'Two Factor Authentication' di menu 'Plugin'
- Klik tombol 'instal (Pastikan memilih yang benar)
- Aktifkan plugin melalui menu 'Plugins'
- Temukan pengaturan seluruh situs di Settings - Two Factor Authentication : temukan pengaturan pengguna di menu entry "Two Factor Auth".

Jika ingin menambahkan bagian ke *front-end* agar pengguna dapat mengonfigurasi pengaturan *two-factor authentication,* maka gunakan shortcode ini : [twofactor_user_settings]

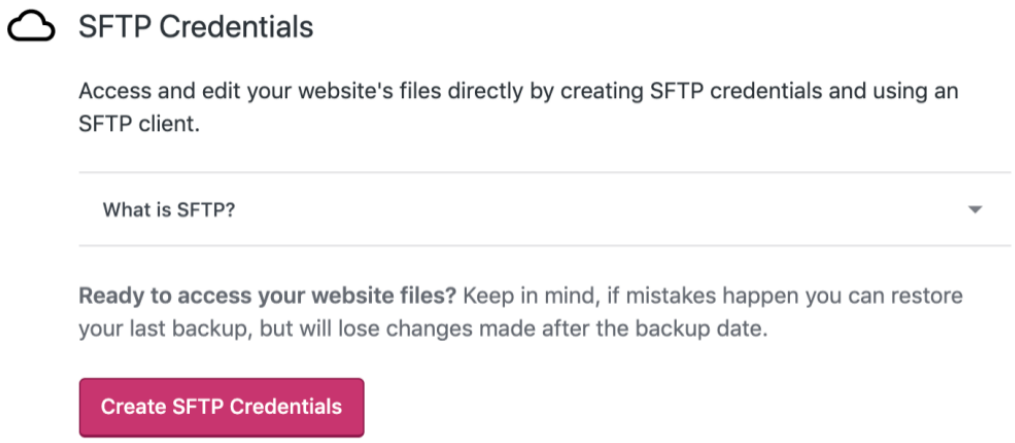
FTP

Saat terhubung ke server disarankan untuk menggunakan enkripsi SFTP (jika web host menyediakan). Jika tidak yakin apakah *host* web menyediakan SFTP atau tidak, perlu untuk menanyakannya pada *web host*.

SFTP (*Secure File Transfer Protocol*) adalah cara untuk mengakses berkas dan folder pada situs web melalui program klien seperti Filezilla pada komputer lokal. Filezilla dapat digunakan pada pengguna Windows, Linux maupun Mac. SFTP dirancang sebagai ekstensi protokol SSH (Secure Shell).

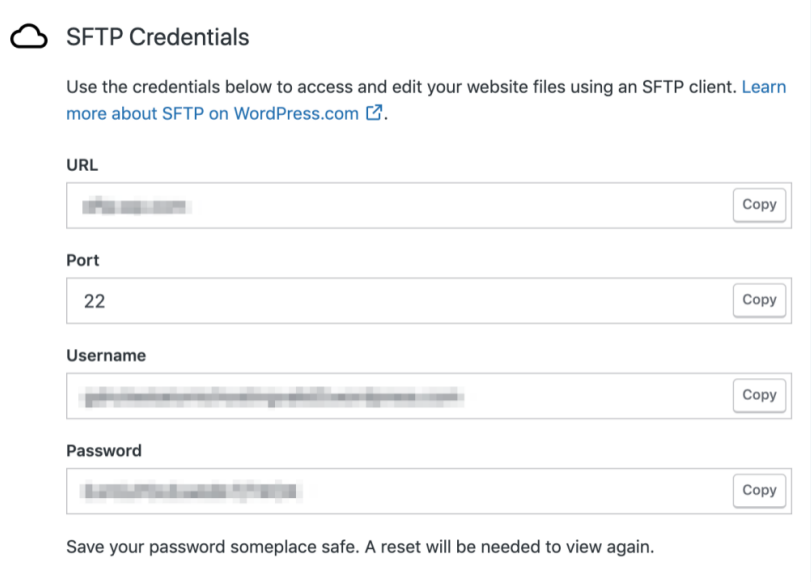
Menggunakan SFTP sama dengan FTP, kecuali *password* dan data lainnya dienkripsi saat dikirimkan antara komputer dan *website*. Ini berarti kata sandi tidak pernah dikirim secara jelas dan tidak dapat disadap oleh penyerang.

WordPress tidak menyiapkan SFTP secara otomatis. Pengguna perlu menekan tombol Aktifkan SFTP untuk mengaktifkan fitur ini.



Simpan informasi URL SFTP, Nomor Port, Nama Pengguna dan Password yang muncul setelah aktifasi fitur SFTP.

Nama pengguna dan kata sandi dihasilkan sistem secara otomatis. Apabila terdapat beberapa situs web maka gunakan beberapa nama pengguna dan password.



Setelah membuat nama pengguna dan password, masukan pada klien SFTP pilihan yang akan digunakan.

PERIZINAN FILE

Beberapa fitur WordPress berasal dari berbagai *file* yang dapat disunting (*writebale*) oleh web server. Namun memberikan izin akses *write* pada *file* merupakan langkah berbahaya, terutama di lingkungan *shared hosting.*

Langkah terbaik adalah dengan mengunci izin *file* sebanyak mungkin dan melakukan pembatasan pada saat mengizinkan akses *write* atau membuat folder tertentu dengan sedikit pembatasan untuk melakukan hal-hal lain, seperti mengunggah *file*.

SKEMA PERIZINAN FILE

Semua *file* harus dimiliki oleh akun pengguna dan dapat diakses. *File* apa pun yang memerlukan akses *write* dari WordPress dapat diakses oleh web server, jika membutuhkan pengaturan *hosting*, berarti *file* tersebut harus dimiliki grup oleh akun pengguna yang digunakan oleh proses web server.

/

Direktori Root WordPress:
Semua file hanya dapat ditulis oleh akun pengguna, kecuali .htaccess. Jika membuat WordPress secara otomatis *rewrite* rules.

/wp-admin/

Area administrasi WordPress:
Semua *file* pada direktori ini hanya dapat diubah oleh administrator atau akun pengguna.

/wp-includes/

Keseluruhan logika aplikasi WordPress:
Semua *file* pada direktori ini hanya dapat diubah oleh akun pengguna.

/wp-content/

Direktori konten website:
Direktori ini disediakan untuk akun pengguna dan proses pada server web.

Di dalam /wp-content/ terdapat:

/wp-content/themes/

Tema file:
Jika ingin mengubah tema bawaan WordPress, maka semua *file* harus mendapatkan akses write oleh proses web server. Jika tidak ingin melakukan edit tema bawaan, akses *write* semua *file* hanya akun pengguna yang memilikinya.

/wp-content/plugins/

Plugin files:
Semua *file* dapat diubah oleh pengguna akun. Direktori lain yang mungkin ada dengan /wp-content/ harus didokumentasikan oleh *plugin* atau tema apapun yang dibutuhkan.

Sistem komputer yang memiliki perbedaan *file* ataupun *direktori* terdapat perijinan yang spesifik untuk siapa dan apa yang akan dibaca (*read*), ditulis (*write*), eksekusi (*execute*). Berikut ini mode perijinan untuk mengaktifkan fungsi-fungsi tertentu :

Dalam hal ini ada tiga kelompok pengguna :

1.User :
Pengguna yang memiliki file

2.Group :
Kelompok pengguna yang telah diberikan akses ke file

3.Public :
Semua orang yang terhubung dengan internet

Perbedaan perijinan yang diberikan kepada masing-masing kelompok pengguna pada file atau direktori :

1.Read **(4) :**
Semua *file* pada direktori ini hanya dapat diubah oleh akun pengguna

2.Write **(2) :**
Kemampuan untuk menulis dan memodifikasi file. Untuk direktori, berarti kemampuan menambah dan menghapis file dalam direktori atau folder

3.e**X**ecute **(1) :**
Kemampuan untuk mengeksekusi file dan menjalankan skrip. Untuk direktori berarti pengguna diijinkan untuk mengakses file yang terdapat di dalamnya.

7	4	4		
user	**group**	**world**		
r+w+x	**r**	**r**		
4+2+1	**4+0+0**	**4+0+0**	**=**	**744**

Contoh Mode Perijinan #

Mode	Str Perijinan	Penjelasan
0477	-r-rwxrwx	pemilik hanya membaca/r (4), group dapat rwx (7)
0677	-rw-rwxrwx	pemilik hanya rw (6), group dapat rwx (7)
0444	-r-r-r-	semua hanya membaca/r (4)

PENGGANTIAN IZIN FILE

Jika memiliki akses *shell* ke server, maka untuk mengubah izin *file* secara rekursif dengan perintah berikut:

Untuk direktori:

```
find /path/to/your/wordpress/install/ -type d -exec chmod 755 {} \;
```

Untuk file:

```
find /path/to/your/wordpress/install/ -type f -exec chmod 644 {} \;
```

Maksud nilai 755 adalah perubahan izin pada direktori, sedangkan nilai 644 merupakan perubahan izin file. Nilai maksimum perubahan izin adalah 777, yang berarti setiap orang, termasuk User, Group, dan Public bisa melakukan tindakan apapun pada file. Nilai minimum akses adalah 444 yang berarti file hanya dapat dibaca.

PEMBARUAN OTOMATIS

Ketika melakulan pembaruan otomatis WordPress, semua operasi *file* dilakukan sebagai pengguna yang memiliki *file* bukan sebagai pengguna server web. Semua *file* diatur ke 0644 dan semua direktori diatur ke 0755, hanya dapat ditulis oleh pengguna dan dapat dibaca oleh semua orang termasuk server web.

KEAMANAN DATABASE

Jika menjalankan beberapa blog di server yang sama, perlu untuk mempertimbangkan penyimpanan di *database* terpisah yang masing-masing dikelola oleh pengguna yang berbeda. Hal ini merupakan langkah yang paling baik dilakukan saat melakukan instalasi WordPress awal yang disebut **strategi penahanan** (*containment strategy*): jika peretas berhasil memecahkan satu instalasi WordPress, dengan adanya strategi ini maka peretas akan lebih sulit untuk mengubah blog yang lainnya.

Jika MySQL dikelola sendiri, disarankan untuk terlebih dahulu memahami konfigurasi MySQL yang diinstal dan fitur yang tidak diperlukan (seperti menerima koneksi TCP jarak jauh) dinonaktifkan.

MEMBATASI PENGGUNA DATABASE

Pembatasan hak istimewa (*privileges*) pengguna *database* perlu dilakukan. Untuk operasi WordPress normal, seperti mengunggah konten pada blog, mengunggah *file* media, mem-*posting* komentar, membuat pengguna WordPress baru dan menginstal *plugin* WordPress, pengguna *database* MySQL hanya memerlukan hak istimewa untuk membaca (*read)* dan menulis (*write*) data ke *database* MySQL; **SELECT, INSERT, UPDATE,** dan **DELETE**.

Oleh karena itu, struktur *database* dan hak administrasi lainnya, seperti **DROP**, **ALTER** dan **GRANT** dapat dicabut. Dengan mencabut hak istimewa tersebut, implementasi kebijakan penahanan meningkat.

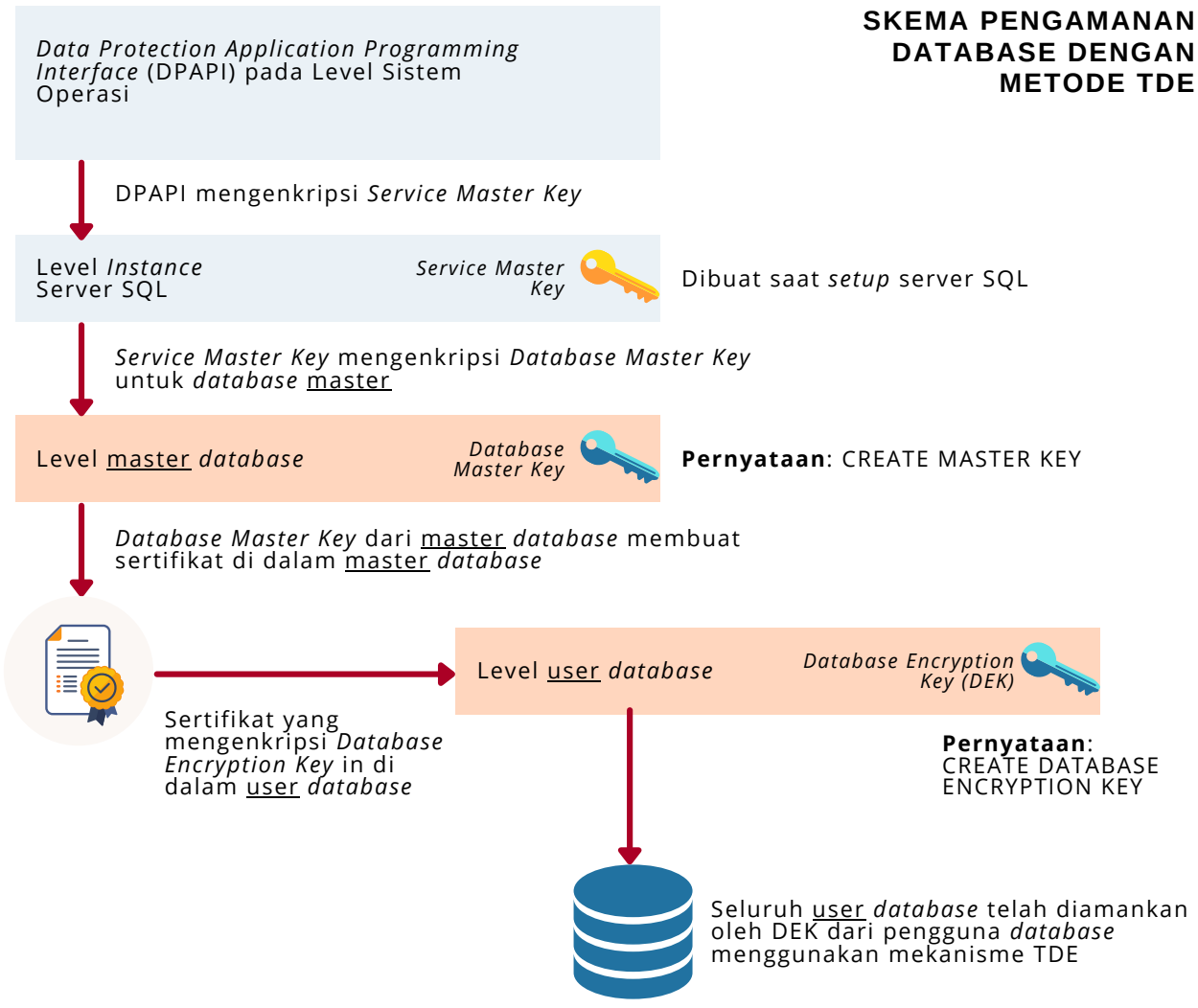
CATATAN:
Beberapa plugin, tema, dan pembaruan WordPress utama mungkin perlu membuat perubahan struktural database, seperti menambahkan tabel baru atau mengubah skema. Dalam kasus seperti itu, sebelum menginstal plugin atau memperbarui perangkat lunak, pengguna database memerlukan izin sementara untuk hak istimewa yang diperlukan.

PERINGATAN:
Percobaan pembaruan tanpa memiliki hak istimewa dapat menyebabkan masalah saat terjadi perubahan skema *database.* Oleh karena itu, TIDAK disarankan untuk mencabut hak istimewa. Jika dirasa perlu dilakukan pencabutan hak istimewa untuk alasan keamanan, maka harus dipastikan ada rencana pencadangan yang solid terlebih dahulu, yaitu pencadangan seluruh *database* reguler yang telah diuji valid dan dapat dengan mudah dipulihkan. Pemutakhiran *database* yang gagal biasanya dapat diselesaikan dengan memulihkannya kembali ke versi lama, memberikan izin yang sesuai, dan kemudian membiarkan WordPress melakukan percobaan pembaruan *database* lagi.

Pemulihan *database* akan mengembalikannya ke versi lama. Layar administrasi WordPress akan mendeteksi versi lama dan memungkinkan dijalankannya perintah SQL yang diperlukan di dalamnya. Sebagian besar peningkatan WordPress tidak mengubah skema, tetapi beberapa dapat mengubah skema, yaitu peningkatan poin utama (3,7 hingga 3,8, misalnya) yang akan mengubah skema. Peningkatan kecil (3,8 hingga 3,8.1) umumnya tidak. Namun demikian, lebih baik menyimpan data cadangan secara teratur dan berkala.

ENKRIPSI DATABASE

Enkripsi *database* merupakan proses yang menggunakan algoritma untuk mengubah data yang disimpan dalam suatu *database* menjadi "kode" yang tidak dapat dipahami tanpa terlebih dahulu didekripsi. Salah satu database yang menyediakan fitur enkripsi adalah MySQL. Sejak versi yang dirilis tahun 2008 MySQL telah menyediakan fitur untuk dapat melakukan enkripsi terhadap *database* yaitu dengan menggunakan **Transparent Data Encryption** (TDE). Fitur ini hanya tersedia di MySQL Server versi 2008 keatas edisi *Data Center* atau *Enterprise*. Enkripsi dilakukan pada level *file* mdf dan log, sehingga jika media penyimpanan *database* hilang atau dicuri, maka database tersebut tidak dapat di *restore* tanpa adanya sertifikat untuk membukanya.



Untuk menggunakan Fitur TDE, dapat melakukan langkah-langkah pada sumber berikut:
https://docs.microsoft.com/en-us/sql/relational-databases/security/encryption/transparent-data-encryption?view=sql-server-ver15

PENGAMANAN WP-ADMIN

Menambahkan perlindungan kata sandi pada sisi server (seperti *BasicAuth*) ke /wp-admin/ merupakan mekanisme penambahan lapisan perlindungan kedua di sekitar area admin blog, layar login, dan *file*. Dengan begitu, penyerang atau bot dipaksa untuk menyerang lapisan perlindungan kedua ini, tidak menuju ke *file* admin yang sebenarnya. Banyak serangan WordPress dilakukan secara mandiri oleh *malware bot*.

Mengamankan direktori /wp-admin/ mungkin juga merusak beberapa fungsionalitas WordPress, seperti handler AJAX di wp-admin/admin-ajax.php.

Serangan paling umum terhadap blog WordPress biasanya terbagi dalam dua kategori.

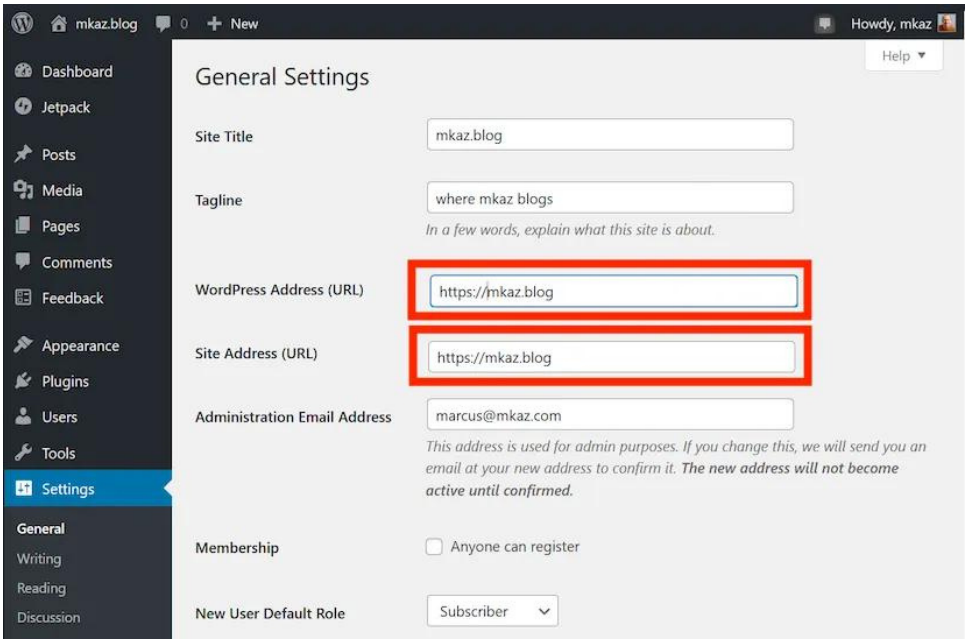
- Mengirim permintaan HTTP yang dibuat khusus ke server target dengan muatan *exploit* khusus untuk kerentanan tertentu, termasuk *plugin* dan perangkat lunak lama/kedaluwarsa.
- Mencoba untuk mendapatkan akses ke blog dengan menggunakan tebakan kata sandi "*brute-force*".

Implementasi akhir dari perlindungan kata sandi "lapisan kedua" ini memerlukan koneksi terenkripsi (HTTPS SSL) untuk administrasi, sehingga semua komunikasi dan data sensitif dienkripsi.

KONFIGURASI HTTPS

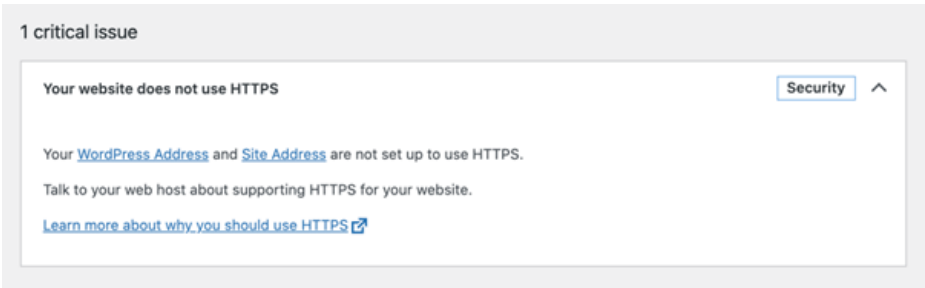
Untuk menerapkan dukungan HTTPS di WordPress, hanya perlu mengatur WordPress dan URL Alamat Situs untuk menggunakan https://.

Konfigurasi dapat dilakukan melalui **Dashboard** CMS Wordpress, yaitu dengan membuka **Settings > General Setting**s.



Jika alamat (URL) masih menggunakan HTTP, maka tambahkan 'S' untuk membuat *website* menjadi https://.

Pemeriksaan status website juga dapat dilakukan dengan membuka tab **Tools > Site health.** Jika masih menggunakan HTTP, maka disarankan untuk menggantinya dengan HTTPS



Sejak versi 5.7, WordPress telah menggunakan fitur yang akan secara otomatis akan mengalihkan URL HTTP menjadi HTTPS jika sertifikat SSL sudah tersedia di server.

REKOMENDASI LANGKAH IMPLEMENTASI HTTPS

Semua situs WordPress disarankan untuk menggunakan HTTPS dengan rincian sebagai berikut:

- Gunakan *website host* yang memiliki reputasi yang baik, biasanya mereka telah menetapkan HTTPS sebagai standar;
- Gunakan sertifikat SSL dari **Let's Encrypt** atau **SmartSSL**, hal ini dikarenakan sertifikat SSL bisa didapatkan secara gratis dan mudah digunakan. Atau dapat menggunakan sertifikat SSL Berbayar; dan
- Sediakan konten statis dari SSL untuk memperbolehkan CDN.

Pemilik website memerlukan untuk meneruskan lalu lintas HTTP ke situs HTTPS. Untuk yang menggunakan Apache server, dapat dilakukan dengan membuat dua entri VirtualHost dengan contoh sebagai berikut:

```
<VirtualHost *:80>
ServerName mkaz.blog
   Redirect / https://mkaz.blog/
</VirtualHost>

<VirtualHost *:443>
ServerName mkaz.blog
DocumentRoot /home/mkaz/sites/mkaz.blog
<Directory /home/mkaz/sites/mkaz.blog>
Options Indexes FollowSymLinks
AllowOverride All
Require all granted
</Directory>

   SSLEngine on
SSLCertificateFile/etc/letsencrypt/live/mkaz.blog/cert.pem
SSLCertificateKeyFile /etc/letsencrypt/live/mkaz.blog/privkey.pem
SSLCertificateChainFile /etc/letsencrypt/live/mkaz.blog/fullchain.pem
IncludeOptional /etc/letsencrypt/options-ssl-apache.conf
</VirtualHost>
```

Sangat tidak disarankan untuk melakukan beberapa hal berikut:

- Melayani situs dari URL HTTPS dan HTTP
- Menggunakan konten campuran, misalkan CSS, JS, atau gambar yang disajikan dari situs yang menggunakan HTTP di halaman HTTPS.

PENGAMANAN WP-INCLUDE

Lapisan perlindungan kedua dapat ditambahkan di mana script umumnya tidak dimaksudkan untuk diakses oleh pengguna mana pun. Salah satu cara untuk melakukannya adalah dengan memblokir scripttersebut menggunakan mod_rewrite di file .htaccess.

Catatan: untuk memastikan *script* di bawah ini tidak ditimpa oleh WordPress, perlu untuk meletakkannya di luar tag # BEGIN WordPress dan # END WordPress di file .htaccess, karena WordPress akan menimpa apa pun yang ada di antara tag-tag ini.

```
# Block the include-only files.
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteBase /
RewriteRule ^wp-admin/includes/ - [F,L]
RewriteRule !^wp-includes/ - [S=3]
RewriteRule ^wp-includes/[^/]+\.php$ - [F,L]
RewriteRule ^wp-includes/js/tinymce/langs/.+\.php - [F,L]
RewriteRule ^wp-includes/theme-compat/ - [F,L]
</IfModule>

# BEGIN WordPress
```

Script di atas tidak akan bekerja dengan baik di Multisite, karena RewriteRule ^wp-includes/[^/]+\.php$ - [F,L] akan mencegah *file* ms-files.php menghasilkan gambar. Dengan menghilangkan baris itu akan memungkinkan kode berfungsi, tetapi keamanannya menjadi lebih rendah.

PENGAMANAN WP-CONFIG.PHP

File wp-config.php dapat dipindahkan ke direktori di atas instalasi WordPress pada perangkat. Dengan begitu, situs yang terinstal di *root* ruang web, wp-config.php dapat disimpan di luar *folder* web-root.

Catatan: Terdapat beberapa pernyataan bahwa memindahkan wp-config.php memberikan keamanan yang minimal dan, jika tidak dilakukan dengan hati-hati, dapat menimbulkan kerentanan serius. Namun beberapa pengguna tidak menyetujui pernyataan tersebut

Perlu diperhatikan bahwa wp-config.php dapat disimpan SATU tingkat direktori di atas instalasi WordPress (tempat wp-includes berada). Pastikan juga bahwa hanya pemilik website (dan server web) yang dapat membaca *file* ini (biasanya berarti izin 400 atau 440). Jika menggunakan server dengan .htaccess, *file* dapat diletakkan (di bagian paling atas) untuk menolak akses siapa pun yang menjelajahinya:

```
<files wp-config.php>order allow,denydeny from all</files>
```

PENONAKTIFAN PENGEDITAN FILE

Dashboard WordPress secara *default* memungkinkan administrator untuk melakukan edit pada *file* PHP, seperti *file* plugin dan tema. Dashboard ini merupakan alat pertama yang akan digunakan penyerang saat berhasil masuk, karena memungkinkan adanya *code execution*. WordPress memiliki konstanta untuk menonaktifkan pengeditan dari Dashboard. Menempatkan baris *script* di bawah ini di wp-config.php sama dengan menghapus kemampuan 'edit_themes', 'edit_plugins' dan 'edit_files' dari semua pengguna:

```
define('DISALLOW_FILE_EDIT', true);
```

Baris script di atas tidak akan mencegah penyerang mengunggah file berbahaya ke website, tetapi mungkin menghentikan beberapa serangan.

PLUGIN

Perlu untuk memastikan bahwa *plugin* selalu diperbarui. Jika tidak menggunakan plugin tertentu, hapus dari sistem.

FIREWALL

Ada banyak *plugin* dan layanan yang dapat bertindak sebagai *firewall* untuk *website*. Beberapa dari *plugin* tersebut bekerja dengan memodifikasi file .htaccess dan membatasi beberapa akses di tingkat Apache, sebelum diproses oleh WordPress. Contoh yang bagus adalah iThemes Security atau All in One WP Security. Beberapa *plugin firewall* bertindak di tingkat WordPress, seperti WordFence dan Shield, dan mencoba memfilter serangan saat WordPress dimuat, tetapi sebelum diproses sepenuhnya.

Selain *plugin firewall*, perimeter dapat juga dengan menginstal WAF (*web firewall*) di server web untuk menyaring konten sebelum diproses oleh WordPress. WAF open-source paling populer adalah ModSecurity.
Website Firewall juga dapat ditambahkan sebagai perantara antara lalu lintas dari internet dan *server hosting*. Semua layanan ini berfungsi sebagai *reverse proxy*, dimana *website firewall* menerima permintaan awal dan merutekannya kembali ke server, menghapus dari semua *malicious request*. Perimeter ini melakukannya dengan memodifikasi catatan DNS, melalui catatan A atau pertukaran DNS penuh, memungkinkan semua lalu lintas melewati jaringan baru terlebih dahulu. Hal ini menyebabkan semua lalu lintas disaring oleh *firewall* sebelum mencapai situs tujuan. Beberapa perusahaan yang menawarkan layanan tersebut, seperti CloudFlare, Sucuri dan Incapsula.

Selain itu, penyedia layanan pihak ketiga ini juga berfungsi sebagai *Content Distribution Network* (CDN) secara default, yang memperkenalkan pengoptimalan kinerja dan jangkauan global.

PLUGIN DENGAN AKSES ‘WRITE’

Jika sebuah *plugin* meminta akses *write* ke *file* dan direktori WordPress yang dimiliki, perlu membaca kode untuk memastikannya sah atau menanyakannya pada seseorang yang dapat dipercaya. Tempat yang memungkinkan untuk diperiksa adalah *Support Forums* dan *IRC Channel*.

PLUGIN CODE EXECUTION

Bagian dari tujuan penguatan keamanan WordPress adalah menahan kerusakan yang terjadi jika terjadi serangan yang berhasil. *Plugin* yang memungkinkan *PHP arbitrer* atau kode lain untuk dieksekusi dari entri dalam database secara efektif memperbesar kemungkinan kerusakan jika terjadi serangan yang berhasil.

Cara untuk menghindari penggunaan *plugin* semacam itu adalah dengan menggunakan *template* halaman khusus yang memanggil fungsi tersebut. Bagian dari keamanan yang diberikan ini hanya aktif ketika pengeditan *file* di dalam WordPress dinonaktifkan.

SECURITY TROUGH OBSCURITY

Security trough obscurity atau Keamanan melalui ketidakjelasan umumnya merupakan strategi utama yang tidak sehat. Namun, terdapat area pada WordPress dimana mengaburkan informasi dilakukan dengan tujuan membantu keamanan. Langkah-langkah tersebut meliputi:

MENGGANTI NAMA AKUN ADMINISTRATIF

Saat membuat akun administratif, perlu untuk menghindari istilah yang mudah ditebak seperti admin atau webmaster sebagai nama pengguna karena mereka biasanya akan diserang terlebih dahulu. Pada instalasi WordPress yang ada, penggantian nama akun yang ada di klien baris perintah MySQL dengan perintah seperti **UPDATE wp_users SET user_login = 'newuser' WHERE user_login = 'admin';**, atau dengan menggunakan *interface* MySQL seperti phpMyAdmin

MENGUBAH TABLE_PREFIX

Banyak serangan injeksi SQL khusus WordPress yang dipublikasikan membuat asumsi bahwa **table_prefix adalah wp_**, defaultnya. Mengubah ini dapat memblokir setidaknya beberapa serangan injeksi SQL

CADANGAN DATA

Disarankan untuk mencadangkan data secara teratur, termasuk *database* MySQL. Integritas data sangat penting untuk pencadangan tepercaya. Dengan mengenkripsi cadangan, menyimpan catatan independen dari *hash* MD5 untuk setiap *file* cadangan, dan/atau menempatkan cadangan pada media *read-only* meningkatkan kewaspadaan bahwa data tidak dirusak. MD5 digunakan pada WordPress karena didukung oleh semua platform.

Strategi pencadangan yang baik dapat mencakup menyimpan satu set *snapshot* secara teratur dari seluruh instalasi WordPress (termasuk *file* inti WordPress dan *database*) di lokasi yang tepercaya. Jika sebuah situs membuat *snapshot* mingguan, strategi seperti itu berarti bahwa jika sebuah situs disusupi pada 1 Mei tetapi penyusupan tidak terdeteksi hingga 12 Mei, pemilik *website* akan memiliki cadangan pra-kompromi yang dapat membantu dalam membangun kembali situs dan bahkan mungkin cadangan pasca-kompromi yang akan membantu menemukan cara bagaimana situs itu disusupi.

LOGGING

Log merupakan fitur terbaik dalam memahami apa yang terjadi dengan *website*, terutama saat dilakukannya forensik. Berlawanan dengan kepercayaan populer, log juga memungkinkan untuk melihat apa yang dilakukan dan oleh siapa dan kapan. Namun, log tidak berisi informasi siapa, nama pengguna, yang masuk, tetapi tetap akan memungkinkan untuk dilakukan identifikasi IP dan waktu dan yang lebih penting, tindakan yang mungkin dilakukan penyerang.

Dengan adanya log ini, serangan-serangan dapat dilihat melalui log – *Cross Site Scripting* (XSS), *Remote File Inclusion* (RFI), *Local File Inclusion* (LFI) dan upaya *Directory Traversal.* Percobaan *brute force* juga dapat diketahui dari log. Ada berbagai contoh dan tutorial yang tersedia untuk membantu memandu untuk melalui proses penguraian dan analisis log mentah.

Di dalam log tersebut juga akan dapat dilihat hal-hal seperti, ketika editor tema dan *plugin* digunakan, ketika seseorang memperbarui *widget* dan ketika dilakukan *posting* dan penambahan halaman, yang merupakan elemen kunci saat melakukan pekerjaan forensik di server web. Ada beberapa *plugin* Keamanan WordPress yang dapat membantu, seperti *Sucuri Auditing tool* atau *Audit Trail plugin.*

Ada dua solusi kunci *open-source* di server web dari perspektif keamanan, hal ini adalah pendekatan berlapis untuk keamanan.

OSSEC dapat berjalan di semua distribusi NIX dan juga akan berjalan di Windows. Ketika dikonfigurasi dengan benar, itu sangat kuat. Idenya adalah mengkorelasikan dan menggabungkan semua log. Perlu dipastikan bahwa konfigurasi diatur untuk menangkap semua access_logs dan error_logs dan jika memiliki beberapa *website* di akun server. Jika ingin memastikan untuk menyaring *noise*, secara *default* akan terlihat banyak *noise* dan mengonfigurasinya agar benar-benar efektif.

MONITORING

Pencegahan saja tidak cukup dan mungkin masih diretas. Hal itu menjadi sebab deteksi/monitoring intrusi sangat penting. Ini akan memungkinkan pemilik web untuk bereaksi lebih cepat, mencari tahu apa yang terjadi dan memulihkan situs yang dimiliki.

MONITORING LOG

Jika pemilik *website* berada di server pribadi khusus atau virtual, di mana memiliki keleluasaan akses *root*, maka pemilik *website* memiliki kemampuan untuk dengan mudah mengonfigurasi berbagai hal sehingga pemilik *website* dapat melihat apa yang terjadi. OSSEC memfasilitasi fitur ini.

MONITORING FILE

Setiap serangan pada sistem selalu meninggalkan jejak. Baik di log atau di sistem *file* (*file* baru, *file* yang dimodifikasi, dll). Jika pemilik *website* menggunakan OSSEC, *file* pada server akan dipantau dan mengingatkan pemilik *website* ketika ada perubahan pada *file*.

TUJUAN

Tujuan dari pelacakan sistem *file* meliputi:

1. Memonitor diubah dan ditambahkan *file;*
2. Mencatat perubahan dan penambahan;
3. Kemampuan untuk mengembalikan perubahan granular;
4. Memberikan perigatan secara otomatis.

PENDEKATAN UMUM

Administrator dapat memantau sistem file melalui teknologi umum seperti:

1. Utilitas sistem;
2. Kontrol revisi;
3. Monitoring tingkat OS/kernel.

SPECIFIC TOOLS

Opsi untuk monitoring sistem *file* meliputi:

1. **diff** – membuat salinan uji bersih situs dan membandingkan dengan produksi
2. **Git –** manajemen kode sumber
3. **inotify dan incron –** layanan monitoring file tingkat kernel OS yang dapat menjalankan perintah pada acara sistem *file*
4. **Watcher –** *Python's Inotify Library*
5. **OSSEC –** *Open Source Host-based Intrusion Detection System* adalah sistem deteksi intrusi berbasis host (HIDS) yang melakukan analisis log, pemeriksaan integritas, pemantauan registri Windows, deteksi rootkit, peringatan berbasis waktu, dan respons aktif. Ini dapat digunakan untuk memantau satu server atau ribuan server dalam mode server/agen.

PERTIMBANGAN

Saat mengonfigurasi strategi monitoring berbasis *file*, ada banyak pertimbangan, termasuk yang berikut ini.

Menjalankan script/layanan monitoring sebagai root
Hal Ini akan mempersulit penyerang untuk menonaktifkan atau memodifikasi solusi monitoring sistem *file* yang dimiliki.

Menonaktifkan monitoring selama pemeliharaan/peningkatan terjadwal
Ini akan mencegah pemberitahuan yang tidak perlu saat pemilik *website* melakukan pemeliharaan rutin.

Memantau tipe file yang dapat dieksekusi
Mungkin cukup aman untuk memantau hanya jenis *file* yang dapat dieksekusi, seperti *file* .php, dll. Memfilter file yang tidak dapat dieksekusi dapat mengurangi entri log dan peringatan yang tidak perlu.

Menggunakan izin sistem file yang ketat
Menurut bagian tentang mengamankan izin dan kepemilikan *file* yang dijelaskan di atas. Secara umum, sedapat mungkin hindari mengizinkan mengeksekusi dan menulis izin.

MONITORING WEB SERVER SECARA EKSTERNAL

Jika penyerang mencoba merusak situs atau menambahkan *malware*, pemilik website dapat mendeteksi perubahan ini dengan menggunakan solusi *web-based integrity monitor*. Caranya cukup mencari di halaman pencarian (Google/Bing/DuckDuckGo/yang biasa digunakan) dengan kata kunci *Web Malware Detection* dan *Remediation*, maka akan muncul daftar penyedia layanan.

HARDENING WORDPRESS

https://wordpress.org/support/article/hardening-wordpress/#logging

DIREKTORAT OPERASI KEAMANAN SIBER

BADAN SIBER DAN SANDI NEGARA

Jalan Harsono RM No.70, Ragunan, Pasar Minggu
Jakarta Selatan 12550, DKI Jakarta, Indonesia.
Tel: +62217805814 Fax: +622178844104
Email: bantuan70@bssn.go.id
info@idsirtii.or.id