

Menjaga Keamanan Sistem

Bagian 1 dari 3 Artikel

Keamanan sistem Linux perlu terus dijaga agar tidak diganggu orang lain, terutama para *cracker* dari Internet. Tidak hanya menjaga keamanan fisik komputer, melainkan keamanan sistem secara keseluruhan. Ini bukan hal yang sulit, tapi juga tidak bisa dianggap remeh.



Jika Anda menjadi seorang *System Administrator*, ada beberapa hal yang harus dilakukan dalam mengamati sistem yang sedang berjalan, sehingga jika ada masalah pemeliharaan atau ada aktivitas yang tidak dikehendaki, Anda bisa melakukan antisipasi pencegahan maupun perbaikan.

Beberapa file yang harus Anda amati, antara lain:

- **/var/log/messages**

File */var/log/messages*, merupakan file yang kali pertama yang harus diamati. Karena dari file ini Anda bisa mengetahui keadaan sistem, akses yang sedang berlangsung, dan mengetahui adanya tindakan yang *tidak biasa* pada sistem, sehingga antisipasi dan pemeliharaan bisa dilakukan lebih dini. Cara mengamatinya adalah sebagai berikut:

```
[root@localhost: ~] # tail -f /var/log/messages
```

- **/var/log/httpd/access_log**

File ini berfungsi melihat aktivitas apache yang sedang berjalan, juga aktivitas user yang sedang mengakses web server, semua bisa dilihat di sini. Untuk mengamatinya, Anda bisa mengetikkan perintah berikut:

```
[root@localhost: ~] # tail -f /var/log/httpd/access_log
```

- **/var/log/squid/access.log**

File */var/log/squid/access.log* bisa digunakan untuk melihat akses user pada server kita. Jika digunakan di warnet (warung Internet) maupun pada ISP kita bisa melihat user di ruang berapa atau user siapa sedang mengakses situs apa saja. Untuk melihatnya, ketikkan perintah berikut:

```
[root@localhost: ~] # tail -f /var/log/squid/access.log
```

- **/var/log/mail**

File di atas berfungsi melakukan pencatatan tentang e-mail yang datang dan terkirim. Dari sini Anda bisa melakukan pelacakan, bila diperlukan untuk memeriksa keberadaan virus komputer dan siapa yang melakukan pengiriman pertama kali. Untuk melihatnya, Anda bisa mengetikkan perintah berikut:

```
[root@localhost: ~] # tail -f /var/log/mail
```

- **/var/log/firewall**

File ini sekadar contoh log buatan penulis, berfungsi untuk melakukan pengamatan keamanan sistem yang sedang berlangsung, sehingga jika ada aktivitas yang mencurigakan, kita bisa mengambil antisipasi yang diperlukan. Untuk melihatnya, Anda bisa mengetikkan perintah sebagai berikut:

```
[root@localhost: ~] # tail -f /var/log/firewall
```

Dari pembahasan di atas, kita bisa mengetahui betapa penting tugas seorang System Administrator dalam menjaga dan mengamankan sistem. Untuk menjaga agar sistem berjalan dengan baik dan antisipasi terhadap user atau penyusup yang tidak diharapkan, kita bisa melakukan pengamanan server. Mari kita bahas bersama.

Tindakan preventif pengamanan server

➔ Pengamanan secara fisik

Beberapa insiden gangguan sistem menunjukkan perlunya pengamanan data secara fisik. Hal ini dikarenakan lebih mudah dari segi aplikasi daripada pembobolan sistem secara *remote* atau jarak jauh. Oleh karena itu, penulis akan menjelaskan beberapa metode pengamanan PC Anda secara fisik.

➔ Penguncian komputer

Pada masa lampau, banyak vendor PC yang menyertakan kunci pada setiap produk yang mereka pasarkan. Ini berguna untuk mengantisipasi dari penyadapan bahkan pencurian data secara fisik. Namun, metode ini sekarang jarang digunakan pada PC rumah tangga (*standalone PC*), beberapa PC *branded* masih menyertakan kunci pada beberapa jenis produk yang dibuat untuk kepentingan perkantoran.

➔ Pengamanan BIOS

BIOS merupakan peranti lunak arus rendah yang berguna untuk melakukan konfigurasi atau manipulasi *hardware* pada PC berbasis x86. MS Windows maupun lilo dari Linux menggunakannya untuk menentukan prosedur melakukan booting pada PC Anda. Anda perlu memproteksi BIOS Anda dengan cara memberi *password* padanya. Cara ini juga berguna untuk mencegah penggunaan *booting up* melalui floppy disk.

Cara ini tidak memberi perlindungan secara maksimal pada PC Anda, namun dapat mempersulit orang lain untuk mengacak-acak PC Anda. Perlu diingat, bahwa seiring dengan kemajuan teknologi, BIOS saat ini menggunakan metode *flashroom*, yang memungkinkan kita untuk mengubah konfigurasi atau bahkan menghapusnya sama sekali. Baik secara fisik, dengan menggabungkan arus positif dan negatif pada baterai atau dengan memberi virus yang dapat mengacak-acak program BIOS yang ada.

➔ Pengamanan Boot Loader

Sistem operasi Linux, sebagaimana UNIX variant lainnya memiliki kelemahan dalam mengamankan bootloader/lilo. Ini dikarenakan untuk berjaga-jaga dalam keadaan

darurat, misalnya jika system administrator lupa password "root", atau melakukan *resqueing system*. Sehingga memungkinkan orang lain untuk ikut serta masuk ke sistem tanpa menyertakan password root, dengan mengetikkan: *linux single* atau *linux -s* pada saat PC dinyalakan dan lilo memberikan prompt. Untuk mengantisipasi-pasinya, Anda perlu memberikan password pada */etc/lilo.conf*, seperti di bawah ini:

```
boot = /dev/hda
vga = 771
read-only
menu-scheme = Wg:kw:Wg:Wg
lba32
prompt
menu-title = "masaji's computer"
restricted
password = 4bl3h
timeout = 80
message = /boot/message
```

```
image = /boot/vmlinuz
label = linux
root = /dev/hda6

image = /boot/vmlinuz-baru
label = linux-baru
root = /dev/hda6
```

Anda perlu menambahkan **restricted** dan **password** pada file */etc/lilo.conf*, seperti terlihat di atas. Lalu jalankan lilo, hasilnya akan tampak seperti berikut ini:

```
[root@localhost: ~] # lilo
Added linux *
Added linux-baru
Added suse
Added memtest86
[root@localhost: ~] #
```

→ Penggunaan xlock dan vlock

Jika Anda sering kali meninggalkan komputer untuk keperluan makan siang, maupun keperluan lain namun tidak ingin melakukan *shutdown* karena sedang tanggung dalam menyelesaikan pekerjaan, Anda bisa menjalankan xlock. Xlock dirancang untuk mengunci konsol X, Anda bisa menggunakannya untuk mengamankan desktop pada saat ingin meninggalkan meja Anda.

Sedangkan bagi Anda yang menggunakan konsol teks, ini biasa yang mengguna-

kan program vi untuk melakukan editing maupun pemrograman, bisa menjalankan vlock. Vlock merupakan program yang berfungsi untuk mengunci beberapa atau semua konsol teks yang terbuka.

→ Pengamanan peranti lunak

Di samping pengamanan secara fisik, Anda juga memerlukan pengamanan sistem melalui berbagai perangkat (*tools*) yang ada pada sistem itu sendiri. Pengamanan ini berguna dalam mengantisipasi penggunaan jaringan yang terhubung dengan PC Anda, lebih-lebih bilamana PC Anda merupakan server utama yang bertugas melayani beragam keperluan dari client, baik lokal maupun Internet.

Berbagai bentuk perlindungan ada di Internet, baik yang ditawarkan sebagai program komersial seperti Xsentry, maupun yang bersifat GPL atau *free*.

→ Pemasangan firewall

Jika Anda mempunyai beberapa buah komputer yang saling terhubung lewat jaringan (ethernet misalnya), dapat menggunakan linux sebagai pintu gerbang (*router/gateway*) untuk menyambungkan semua komputer Anda menuju Internet.

Router dan gateway sendiri sebenarnya secara teori mempunyai filosofi arti yang berbeda, gateway sebenarnya mengacu pada alat yang difungsikan untuk menjembatani dua buah jaringan yang mempunyai topologi berbeda, berbeda subnet, dan sebagainya, sedangkan router untuk mengatur pengalamanan paket paket data dalam jaringan yang berbeda sehingga komunikasi dapat terlaksana.

Akan tetapi, dalam kenyataan sehari hari router dan gateway sering kali hanya ditangani oleh sebuah alat saja. Hal inilah yang menyebabkan router selalu diidentikkan dengan gateway, demikian pula sebaliknya.

Pada saat PC kita bisa berhubungan satu sama lain, maka akan kita temui satu persoalan baru, yaitu bagaimana agar kita tidak kedatangan "tamu tak diundang". Untuk itulah kita buat firewall.

Jika server Anda bukan merupakan layanan untuk publik, tentukan dari host/IP mana saja yang diizinkan untuk mengakses ke server Anda, kemudian lakukan DENY/

DROP/REJECT terhadap paket-paket yang bukan dari host/IP yang Anda izinkan, gunakan *ipfwadm/ipchains/iptables/ipfw*, atau *ipf* untuk melakukan filtering/firewalling.

Firewall adalah suatu cara untuk membatasi informasi yang dibolehkan masuk dan keluar dari jaringan lokal Anda. Umumnya host firewall terhubung ke Internet dan LAN lokal Anda, dan akses LAN Anda ke Internet hanya melalui firewall. Dengan demikian, firewall dapat mengendalikan apa yang diterima dan dikirim dari Internet dan LAN Anda.

Firewall adalah teknik yang sangat berguna dan penting dalam mengamankan jaringan Anda. Penting untuk menyadari bahwa Anda tidak boleh pernah berpikir bahwa dengan memiliki firewall, Anda tidak perlu mengamankan mesin-mesin di baliknya. Ini kesalahan fatal. Periksa Firewall-HOWTO yang sangat bagus di arsip terbaru *sunsite* untuk informasi mengenai firewall dan Linux. <http://sunsite.unc.edu/mdw/HOWTO/Firewall-HOWTO.html>.

→ Setting firewall

Setting firewall personal ini sekilas kelihatan kalau ruwet, namun sebenarnya tidak begitu. Di sini disertakan *maquerading*, karena bilamana ada rekan kita yang membawa laptop ingin "*nebang*" jaringan Internet, ataupun berbagi file dan data, maka kita akan membukanya satu per satu. Selain itu, sebagai jaga-jaga kalau ada rekan kita yang usil "*ngerjain*" PC kita, karena "sesama bis kota dilarang saling mendahului".

→ Pemasangan firewall dengan Iptables

Penggunaan IPTABLES ini diperuntukkan bagi yang menggunakan kernel 2.4.x, program IPTABLES ini lebih lengkap dari segi fasilitasnya dan juga lebih "aman" dibandingkan ipchains. Disarankan Anda menggunakan IPTABLES versi 1.2.5 ke atas untuk amannya.

→ Penggunaan Portsentry

Portsentry dirancang khusus untuk memperlambat proses penjejakan (*scanning*) yang biasa dilakukan oleh user, baik

user eksternal maupun user internal sendiri. Di dalam buku ini sudah disertakan CD, yang salah satunya berisi program Portsentry.

Cara kerja program ini pada dasarnya akan melakukan indikasi terhadap penjajakan (scanning) melalui syslog. Jika ada, sasaran akan secara otomatis diarahkan kepada */etc/hosts.deny*. Portsentry akan melakukan konfigurasi ulang terhadap host lokal terhadap route penjajak (scanner), sehingga akan mengakibatkan sistem menjadi "menghilang". Selanjutnya, host lokal akan melakukan blokir melalui filter yang ada.

Pengamanan lokal

Setelah kita membahas bersama masalah penggunaan firewall dan Portsentry dalam mengantisipasi penyusupan ataupun gangguan terhadap sistem, sekarang kita menginjak pada pengamanan sistem internal. Pengamanan ini tidak kalah pentingnya dibandingkan penggunaan firewall. Keduanya, jika dipadukan akan

memberikan hasil yang optimal dalam melindungi sistem dari orang-orang yang tidak berhak. Berikut ini penjelasan sederhana mengenai pengamanan lokal jaringan.

➔ Pembuatan rekening baru

Anda harus memastikan bahwa Anda memberikan *account* kepada orang yang tepat sesuai kebutuhan dan tugas-tugasnya. Beberapa aturan yang penting bagi orang yang boleh mengakses sistem Anda, sebagai berikut:


- Beri mereka fasilitas minimal sepanjang yang dibutuhkan saja.
- Perhatikan jika mereka mencoba untuk mengakses area yang bukan hak mereka.
- Hapus rekening mereka seketika mereka tidak lagi membutuhkan akses.

➔ Pengamanan root

Root merupakan account yang paling dicari oleh para perusak sistem (*cracker*), beberapa hal yang perlu Anda pastikan

dalam mengamankan account root, adalah sebagai berikut:

- a. Jangan sekali-sekali login sebagai root, jika tidak sangat perlu.
- b. Jika terpaksa ingin menggunakan root, login-lah sebagai user biasa kemudian gunakan perintah *su* (*substitute user*).
- c. Jangan sekali-sekali menggunakan seperangkat utilitas, seperti *rlogin/rsh/rexec* (disebut utilitas *r*) sebagai root. Mereka menjadi sasaran banyak serangan dan sangat berbahaya bila dijalankan sebagai root. Jangan membuat file *.rhosts* untuk root.
- d. Jangan pernah menggunakan *."*, yang berarti direktori saat ini dalam penyertaan path.

Bagaimana mengamankan sistem file, password, enkripsi dan PAM (*Pluggable Authentication Modules*), PGP (*Pretty Good Privacy*), dan SSL (*Secure Sockets Layer*)? Baca artikel selanjutnya di *InfoLINUX* Juli 2003. 

R. Kresno Aji (*masaji@ai.co.id*)

IKLAN

Menjaga Keamanan Sistem

Bagian 2 dari 3 Artikel

Untuk menjaga sistem Linux Anda aman, perhatikan pemakaian *user root*, akses terhadap file, pengaturan *login*, *password*, dan enkripsinya.



Beberapa poin ini merupakan lanjutan petunjuk pengamanan *root* yang telah disinggung di bagian pertama.

- ➔ Batasi penggunaan konsol untuk *login* sebagai *root* hanya pada konsol 1 dan 2 dengan cara menghapus *tty3* dan *tty4*. Untuk membatasinya, Anda bisa melakukan editing pada */etc/securetty*, seperti tampak pada gambar berikut:

```
#
# This file contains the device names of
# tty lines (one per line,
# without leading /dev/) on which root is
# allowed to login.
#
tty1
tty2
# for devfs:
vc/1
vc/2
```

- ➔ Batasi waktu penggunaan *account root*, ini berguna jika pada suatu saat Anda lupa melakukan *logout* dari *account root*. Anda perlu membatasi waktu *time out* pada saat *account root* nonaktif. Cara membatasinya waktunya sebagai berikut:

- Edit file */root/.bashrc*
- Tambahkan variabel di bawah ini:


```
HISTFILESIZE=100
HISTSIZ=30
TMOUT=900
```

Variabel *TMOUT* berfungsi untuk menghitung waktu pada saat *user root* sedang tidak aktif. Lamanya dihitung berdasarkan detik (*second*). Jadi jika Anda menginginkan *account root* untuk otomatis *logout* dari *shell* selma 15 menit tidak ada aktivitas, Anda bisa melakukan setting 60 detik x 15 menit = 900. *HISTSIZ*

berguna untuk membatasi *history* pengetikan yang dilakukan oleh *root* sampai dengan 30 baris. Sedangkan *HISTFILESIZE* berfungsi untuk membatasi ukuran file *.bash_history* sampai dengan 100 byte saja selanjutnya akan dihapus. Dengan demikian, akan mengecoh orang lain yang ingin melakukan pengintipan di *.bash_history*. Anda juga bisa memasangnya di */etc/profile*, untuk menerapkannya terhadap seluruh *user*. Untuk melihat efeknya, Anda harus *logout* terlebih dahulu. Kemudian *login* kembali sebagai *user root*.

- ➔ Untuk lebih mengamankan *account root*, edit file */root/.bashrc*, dan isikan variabel berikut:

```
cat /dev/null > /root/.bash_history
```

Variabel di atas akan mengakibatkan sistem secara otomatis akan mengosongkan *.bash_history* pada saat *user root* melakukan *login*. Variabel ini bisa juga Anda masukkan di file *.bashrc* pada masing-masing *home user* untuk keamanan mereka sendiri.

- ➔ Anda juga bisa memasukkan perintah pengosongan *history* melalui *crontab*, seperti berikut ini:

```
[root@localhost: ~] # crontab -e
[Insert]
59 * * * * cat /dev/null > /root/.bash_history
[Esc]
:wq [Enter]
```

Perintah di atas akan mengakibatkan sistem mengosongkan file *.bash_history* pada direktori */root* setiap satu jam sekali.

Pengamanan file dan sistem File

Mengamankan file dan sistem file sangat penting, karena akan sangat berarti bagi penyusup untuk menggunakan file yang ada demi kepentingan penyusup. Misalnya, penggunaan *gcc* untuk kompilasi program, akan sangat berbahaya jika dilakukan oleh *user* yang tidak bisa dipercaya. Beberapa menit persiapan dan perencanaan sebelum menaruh sistem Anda *online* dapat membantu melindungi sistem Anda, dan data yang disimpan. Berikut ini beberapa tips untuk mengamankan file dan sistem file Anda.

- Tidak ada alasan untuk menjalankan program *SUID/SGID* dari *home user*. Gunakan opsi "*nosuid*" dalam */etc/fstab* untuk partisi yang dapat ditulis oleh orang selain *root*. Anda bisa juga menambahkan "*nodev*" dan "*noexec*" di partisi *\$HOME/user*, juga di */var*, yang melarang eksekusi program, dan penciptaan *device* karakter atau blok, yang sebenarnya tidak perlu. Misalnya untuk mengamankan direktori */tmp* dan direktori */home*, contohnya sebagai berikut:

<i>/dev/hda7</i>	<i>/tmp</i>	<i>ext2</i>
<i>nosuid,defaults</i>	1 1	
<i>/dev/hdc2</i>	<i>/home</i>	<i>ext2</i>
<i>nodev,noexec,defaults</i>	1 2	

- Konfigurasi *umask* penciptaan file *user* sistem Anda perlu dijaga seketat mungkin. Setting yang biasa digunakan adalah 022, 033, dan yang paling ketat adalah 077, dan ditambahkan ke */etc/profile*.
- Set limit sistem file. Anda dapat mengendalikan limit tiap pemakai menggunakan module *PAM* dan */etc/security/limits.conf*. Sebagai contoh,

limit untuk kelompok “users” mungkin tampak sebagai berikut:

@users	hard core	0
@users	hard nproc	50
@users	hard rss	5000

Perintah ini berarti melarang penciptaan file *core*, membatasi jumlah proses hingga 50, dan membatasi penggunaan memory tiap user hingga 5M. File */var/log/wtmp* dan */var/run/utmp* berisi catatan login seluruh pemakai sistem Anda. Integritasnya harus dipelihara karena dapat digunakan untuk menentukan kapan dan dari mana seorang pemakai (atau penyusup potensial) memasuki sistem Anda. File-file ini harus memiliki permisi 644, tanpa mempengaruhi operasi sistem normal.

- Jika Anda mengekspor sistem file menggunakan NFS, pastikan mengonfigurasi */etc/exports* dengan akses yang seketat mungkin. Artinya tidak menggunakan *wildcard*, tidak membolehkan root akses menulis, dan melakukan *mount read-only* jika mungkin.
- *Bit immutable* dapat digunakan untuk mencegah penghapusan atau penimpahan sebuah file yang harus dilindungi tanpa sengaja. Juga dapat mencegah seseorang menciptakan *link* simbolik ke file ini, yang telah merupakan sumber penyerangan melibatkan penghapusan */etc/passwd* atau */etc/shadow*. Misalkan melindungi kedua dile tersebut di atas, caranya adalah sebagai berikut:

```
masaji:/etc # chattr +ua shadow
masaji:/etc # chattr +ua passwd
```

Perintah di atas akan mengakibatkan file *shadow* dan *passwd* tidak akan bisa dihapus dan hanya bisa ditambahi saja (*append*).

- File-file SUID dan SGID pada sistem Anda adalah risiko keamanan potensial, dan harus diawasi dengan baik. Oleh karena program-program ini memberi izin khusus bagi pemakai yang mengeksekusinya, maka perlu dipastikan bahwa program yang tidak aman tidak diinstalasi. Trik favorit

cracker adalah mengeksploitasi program SUID “root”, lalu meninggalkan program SUID sebagai *backdoor* untuk masuk di saat lain, meski lubang yang asli telah ditutup. Carilah seluruh program SUID/SGID di sistem Anda, dan catatlah, sehingga Anda mengerti setiap perubahan yang dapat mengindikasikan penyusup potensial. Gunakan perintah berikut untuk mencari seluruh program SUID/SGID di sistem Anda.

- Perketat sekuriti Anda dengan menyunting file */etc/login.defs*, dengan melakukan editing pada bagian ini:

```
# Password aging controls:
#
# PASS_MAX_DAYS Maximum number
# of days a password may be used.
# PASS_MIN_DAYS Minimum number of
# days allowed between password changes.
# PASS_MIN_LEN Minimum acceptable
# password length.
# PASS_WARN_AGE Number of days
# warning given before a password expires.
#
PASS_MAX_DAYS 30
PASS_MIN_DAYS 0
PASS_MIN_LEN 8
PASS_WARN_AGE 28
```

- Berikan akses seminim mungkin bagi daemon atau program yang berjalan dan bila mungkin masukkan program tersebut ke dalam *chrooted-jail*, tentunya akses yang Anda berikan jangan sampai “menjerat” program/daemon tersebut hingga mati.
- Monitor selalu proses (‘ps ax’), log (‘/var/log/’), dan file file yang terbuka (‘ls -l’).
- Perhatikan pula *socket* yang terbuka, dan *socket connections*, dengan *netstat*, contoh:


```
[root@hartx /root]# netstat -atn
```
- Berikan perhatian ekstra jika Anda menjalankan layanan ftp publik (anonymous ftp) terutama sekali jika Anda menggunakan Wu-ftp, juga pada DNS Server Anda jika Anda menggunakan BIND, karena kedua service ini paling rentan terhadap

masalah keamanan (terbukti sering terjadi “tambal sulam” pada dua program ini).

- Akhirnya, sebelum mengubah perizinan di sembarang sistem file, pastikan Anda paham apa yang Anda lakukan. Jangan pernah mengubah permisi suatu file hanya karena ini tampaknya merupakan cara termudah menyelesaikan sesuatu. Selalu tentukan mengapa file memiliki permisi tersebut sebelum mengubahnya.

Pemeliharaan System Account Data

Sangat penting bahwa informasi yang berasal dari *syslog* belum diganggu. Membuat file dalam */var/log* dapat dibaca dan ditulis oleh sejumlah pemakai terbatas adalah awal yang baik. Yakinkan untuk memperhatikan apa yang ditulis di sana, khususnya dalam fasilitas ‘*auth*’. Banyaknya kegagalan *login*, sebagai contoh, dapat mengindikasikan usaha *break-in*.

Ke mana untuk melihat file log Anda tergantung pada distribusi Anda. Dalam sistem Linux yang sesuai dengan “Linux Filesystem Standard”, seperti Red Hat, Anda ingin melihat ke */var/log* dan memeriksa pesan-pesan, *mail.log* dan lainnya.

Anda dapat menemukan di mana distribusi Anda mencatat dengan melihat pada file */etc/syslog.conf*. Ini file yang memberitahu *syslogd* (*the system logging daemon*) di mana mencatat berbagai pesan. Anda mungkin ingin mengkonfigurasi *script log-rotating* Anda atau daemon untuk menjaga log lebih panjang sehingga Anda memiliki waktu untuk memeriksanya. Lihat paket ‘*logrotate*’ dalam sistem Linux versi terbaru.

Jika file log Anda telah diganggu, lihat bila Anda dapat menentukan kapan terjadinya, dan hal-hal apa yang diganggu. Apakah ada periode waktu yang tidak dapat dihitung? Periksa *tape back-up* (jika Anda punya) untuk file log yang tidak terganggu adalah ide yang baik.

File log umumnya dimodifikasi oleh penyusup dalam rangka menutup

jejaknya, tetapi mereka harus juga memeriksa kejadian-kejadian aneh. Anda mungkin memperhatikan penyusup berusaha memperoleh jalan masuk, atau mengeksploitasi program dalam rangka memperoleh rekening root. Anda mungkin melihat masukan log sebelum penyusup memiliki waktu memodifikasi mereka.

Anda harus juga yakin untuk memisahkan fasilitas *'auth'* dari data log lain, termasuk usaha untuk mengganti pemakai menggunakan *'su'*, usaha login, dan informasi akuntansi pemakai lainnya.

Jika mungkin, konfigurasi syslog untuk mengirim salinan data yang paling penting ke sistem yang aman. Hal ini akan mencegah penyusup menutupi jejaknya dengan menghapus usaha login */su/ftp/etc*. Lihat *syslog.conf* man page, dan acu pilihan *'@'*.

Akhirnya, file log kurang berguna ketika tak seorang pun membacanya. Lihatlah log file Anda sewaktu-waktu, dan kenali tampaknya untuk hari normal. Dengan mengetahui hal ini dapat membantu mengenali hal-hal yang tidak biasa.

Perizinan file

Seorang sistem administrator perlu memastikan bahwa file-file pada sistem tidak terbuka untuk pengeditan oleh pemakai dan grup yang tidak seharusnya melakukan pemeliharaan sistem semacam itu.

UNIX membedakan kendali akses pada file dan direktori berdasarkan tiga karakteristik: pemilik (*owner*), grup, dan yang lain (*other*). Selalu terdapat satu pemilik, sejumlah anggota grup, dan setiap orang lain. Penjelasananya demikian:

Read (Baca):

- Mampu melihat isi file.
- Mampu membaca direktori.

Write (Menulis):

- Mampu menambah atau mengubah file.
- Mampu menghapus atau memindah file dalam sebuah direktori.

Execute (Eksekusi):

- Mampu menjalankan program biner atau *script shell*.
- Mampu mencari dalam sebuah direktori, dikombinasikan dengan permissi *read*.

Menyimpan atribut teks: (untuk direktori) Bit sticky juga memiliki arti lain ketika diaplikasikan pada direktori. Jika bit sticky diset pada direktori, maka seorang pemakai hanya boleh menghapus file yang dimiliki atau diberi ijin menulis secara eksplisit, walaupun ia memiliki akses ke direktori. Hal ini dirancang untuk direktori seperti */tmp*, yang bersifat *world-writable*, tetapi tidak diinginkan setiap pemakai dapat menghapus file sesukanya. Bit sticky dilihat sebagai sebuah *'t'* dalam daftar direktori.

Atribut SUID: (untuk file) Atribut ini menggambarkan permissi set ID pemakai atas file. Ketika mode akses permissi set ID diset dalam permissi pemilik, dan file adalah eksekutabel, proses yang menjalankannya diberi izin akses kepada sumber daya sistem berdasarkan pemakai yang membuat proses. Inilah penyebab eksploitasi *'buffer overflow'*.

Atribut SGID: (untuk file) Jika diset dalam permissi grup, bit ini mengendalikan status "set group id" file. Ia berlaku serupa dengan SUID, kecuali grup terpengaruh. File harus eksekutabel agar dapat berlaku.

Atribut SGID: (untuk direktori) Jika Anda menset bit SGID pada direktori (dengan *"chmod g+s direktori"*), file yang tercipta di direktori akan memiliki grup yang sama dengan grup direktori.

Pengamanan password dan enkripsi

Salah satu fitur keamanan yang penting yang digunakan saat ini adalah *password*. Penting bagi Anda dan seluruh pemakai *server* Anda untuk memiliki password yang aman dan tidak dapat diterka. Kebanyakan distribusi Linux terbaru menyertakan program *'passwd'* yang tidak membolehkan Anda menset password yang mudah diterka. Pastikan

program *passwd* Anda terbaru dan memiliki fitur ini.

Kebanyakan *unixies* (dan Linux bukanlah perkecualian) utamanya menggunakan algoritma enkripsi satu arah (*one-way*), disebut DES (*Data Encryption Standard*) untuk mengenkripsi password Anda. Password terenkripsi ini kemudian disimpan di */etc/passwd* atau di */etc/shadow*. Ketika Anda berusaha login, apapun yang Anda ketikkan dienkripsi dibandingkan dengan masukkan dalam file yang menyimpan password Anda. Jika cocok, pastilah password-nya sama, dan Anda dibolehkan mengakses. Meskipun DES merupakan algoritma enkripsi dua arah (Anda dapat meng-*code* dan men-*decode* pesan, dengan memberi kunci yang tepat), varian yang digunakan kebanyakan *unixies* adalah satu arah. Artinya tidak mungkin membalik enkripsi untuk memperoleh password dari isi */etc/passwd* (atau */etc/shadow*).

Serangan *brute force*, seperti *"Crack"* atau *"John the Ripper"* sering dapat digunakan untuk menerka password meski password Anda cukup acak. Modul PAM memungkinkan Anda menggunakan rutin enkripsi yang berbeda dengan password Anda (MD5 atau sejenisnya).

Shadow passwords

Shadow password adalah suatu cara menjaga password terenkripsi Anda dari pemakai normal. Normalnya, password terenkripsi ini disimpan di file */etc/passwd* dapat dibaca semua pemakai. Mereka lalu dapat menjalankan program penerka password dan berusaha menentukan password-nya. *Shadow password* menyimpan informasi ini ke file */etc/shadow* yang hanya dapat dibaca oleh pemakai yang berhak. Dalam rangka menjalankan *shadow password* Anda perlu memastikan bahwa seluruh utilitas Anda yang perlu mengakses informasi password dikompilasi ulang untuk mendukungnya. PAM juga membolehkan Anda untuk hanya memasukkan modul *shadow* dan tidak perlu mengkompilasi ulang eksekutabel. Anda dapat mengacu pada *Shadow-*

Password HOWTO untuk informasi lebih lanjut jika perlu. Contohnya sebagai berikut:

```
tamu:x:502:100:Hanya Tamu:/home/tamu:/bin/bash
hartx:x:503:100:Agus Hartanto:/home/hartx:/bin/bash
```

Pada contoh di atas, password dari user hartx dan tamu hanya disimboliskan sebagai **x**, karena yang sesungguhnya ada dan tersimpan di file `/etc/shadow`. Mari kita lihat isi password yang sebenarnya:

```
tamu:AdtjuUotiABk:11625:0:99999:7:0::
hartx:SbTexoldqe4lQ:11631:0:99999:7:0::
```

Terlihat bahwa dengan penggunaan shadow, password dari user yang bersangkutan sudah dienkripsi sedemikian rupa, sehingga hanya dengan super komputer saja password tersebut dapat diuraikan. Dan kalau ingin lebih cepat, tentu saja melalui metode pendekatan personal).

SSH (Secure Shell), stelnets


SSH dan stelnets adalah program yang memungkinkan Anda untuk login ke sistem remote dan memiliki koneksi yang terenkripsi. SSH adalah paket program yang digunakan sebagai pengganti yang aman untuk rlogin, rsh dan rcp. Ia menggunakan *public-key cryptography* untuk mengenkripsi komunikasi antara dua host, demikian pula untuk autentikasi pemakai. Ia dapat digunakan untuk login secara aman ke remote host atau menyalin data antar-host, sementara mencegah *man-in-the-middle attacks* (pembajakan sesi) dan DNS spoofing. Ia akan melakukan kompresi data pada koneksi Anda, dan komunikasi X11 yang aman antar-host. SSH homepage dapat dijumpai di <http://www.cs.hut.fi/ssh>.

Anda dapat pula menggunakan SSH dari stasiun kerja Windows Anda ke server SSH Linux. Terdapat beberapa implementasi client Windows yang tersedia gratis, termasuk satu di <http://guardian.htu.tuwien.ac.at/therapy/ssh/>

dan juga implementasi komersial dari DataFellows, di <http://www.datafellows.com>.

SSLeay adalah implementasi bebas protokol *Secure Sockets Layer Netscape*, termasuk beberapa aplikasi, seperti Secure telnet, modul untuk Apache, beberapa database, dan juga beberapa algoritma termasuk DES, IDEA dan Blowfish.

Dengan menggunakan pustaka ini, pengganti secure telnet telah diciptakan yang melakukan enkripsi pada koneksi telnet. Tidak seperti SSH, stelnets menggunakan SSL, protokol Secure Sockets Layer yang dikembangkan Netscape. Anda dapat menjumpai Secure telnet dan Secure FTP dengan melihat dulu SSLeay FAQ, tersedia di <http://www.psy.uq.oz.au>.

Edisi mendatang mengulas PAM (*Pluggable Authentication Modules*), PGP (*Pretty Good Privacy*) dan Public Key Cryptography, SSL (*Secure Sockets Layer*), Tripwire, dan Trojan Horse. 

R. Kresno Aji (masaji@ai.co.id)

IKLAN

Menjaga Keamanan Sistem

Bagian 3 dari 3 Artikel

Cracker merupakan ancaman yang biasanya datang dari Internet. Aplikasi atau *server* yang terhubung ke Internet harus mendapat perhatian khusus, misalnya dengan menerapkan PGP, S-HTTP, dan SSL.

Sebelum mengulas cara pengamanan dari gangguan eksternal, awal dari bagian tiga ini masih terkait dengan pengamanan internal, yaitu tentang PAM. Pengamanan internal yang telah dibahas pada edisi yang lalu antara lain pengamanan file, *account*, dan *password*.

PAM-Pluggable Authentication Modules

PAM memungkinkan Anda mengubah secara *on the fly* metode otentikasi Anda, dan menyembunyikan seluruh metode otentikasi lokal tanpa perlu mengompilasi ulang *biner* Anda. Konfigurasi PAM adalah di luar lingkup dokumen ini, tetapi pastikan untuk melihat situs PAM untuk informasi lebih lanjut (<http://www.kernel.org/pub/linux/libs/pam>).

Beberapa hal yang dapat Anda lakukan dengan PAM:

- Menggunakan enkripsi non-DES untuk password Anda. Ini membuatnya sulit untuk didekodekan secara *brute force*.
- Mengeset batasan sumber daya pada seluruh pemakai Anda, sehingga mereka tidak dapat melakukan serangan *Denial of Service* (pembatasan jumlah proses, jumlah memori, dan sebagainya)
- Memungkinkan *shadow password* secara *on the fly*.
- Membolehkan pemakai tertentu untuk login hanya pada waktu tertentu dari tempat tertentu.

Dalam beberapa jam setelah instalasi dan konfigurasi sistem Anda, Anda dapat mencegah banyak serangan sebelum mereka terjadi. Sebagai contoh, menggunakan PAM untuk meniadakan pemakain secara *system-wide* terhadap

file *dot-rhosts* dalam direktori home pemakai dengan menambahkan baris-baris berikut ke `/etc/pam.d/login`:

```
login auth required pam_rhosts_auth.so
no_rhosts
```

Pengamanan eksternal

Pengamanan eksternal sangat diperlukan, khususnya dalam transaksi perdagangan maupun pengiriman informasi pada umumnya. Ada beberapa macam pengamanan eksternal, di antaranya yang akan penulis paparkan berikut ini.

PGP dan Public Key Cryptography

Public Key Cryptography, seperti yang digunakan untuk PGP (*Pretty Good Privacy*), melibatkan kriptografi yang menggunakan satu kunci untuk enkripsi, dan satu kunci untuk dekripsi. Secara tradisional, kriptografi menggunakan kunci yang sama untuk enkripsi dan dekripsi. "Kunci pribadi" ini harus diketahui oleh kedua pihak dan ditransfer dari satu ke lainnya secara aman.

Enkripsi kunci publik secara aman mentransmisi kunci yang diperlukan untuk enkripsi dengan menggunakan dua buah kunci berbeda, kunci pribadi dan kunci publik. Kunci publik setiap orang tersedia bagi semua orang untuk melakukan enkripsi, sementara pada saat yang sama setiap orang menjaga kunci pribadinya untuk mendekripsi pesan terenkripsi dengan kunci publik yang tepat.

Terdapat keuntungan *public key* dan *private key cryptography*, dan Anda dapat membaca tentang perbedaan-perbedaan ini dalam RSA Cryptography FAQ (dijelaskan kembali pada akhir bagian ini).

PGP didukung dengan baik oleh Linux. Versi 2.6.3 dan 5.2 dikenal bekerja dengan baik. Untuk pengenalan tentang



PGP dan bagaimana menggunakannya, silakan lihat PGP FAQ <http://www.pgp.com/service/export/faq/55faq.cgi>. Pastikan menggunakan versi yang dapat digunakan di negara Anda, berkaitan dengan pembatasan ekspor oleh pemerintah AS. Enkripsi kuat dianggap sebuah senjata militer, dan terlarang untuk ditransfer dalam bentuk elektronik ke luar negeri.

Terdapat pula panduan langkah-demi-langkah untuk mengonfigurasi PGP pada Linux di <http://mercury.chem.pitt.edu/~angel/LinuxFocus/English/November1997/article7.html>. Ditulis untuk versi internasional PGP, tetapi dapat diterapkan secara mudah ke versi AS. Anda mungkin butuh patch bagi versi terbaru Linux, yang tersedia di <ftp://sunsite.unc.edu/pub/Linux/apps/crypto>.

Informasi lebih jauh tentang cryptography dapat dijumpai dalam RSA Cryptography FAQ, tersedia di <http://www.rsa.com/rsalabs/newfaq>. Di sini Anda akan menjumpai informasi mengenai istilah seperti "*Diffie-Hellman*", "*public-key cryptography*", "*Digital Certificates*", dan sebagainya.

S-HTTP dan S/MIME

Seringkali pemakai bertanya mengenai perbedaan-perbedaan antara berbagai protokol keamanan, dan bagaimana menggunakannya. Meski ini bukan dokumen enkripsi, merupakan ide yang baik untuk menjelaskan secara singkat setiap protokol dan di mana mencari informasi yang lebih banyak.

- S-HTTP: *Secure-HyperText Transfer Protocol* adalah protokol lain dari HTTP yang memberikan pelayanan keamanan di Internet. Ia dirancang untuk memberikan *confidentiality*, *authentic-*

ity, *integrity*, dan *non-repudiability* (tidak dapat dianggap sebagai orang lain), dan mendukung banyak mekanisme manajemen kunci dan algoritma kriptografi melalui pilihan negosiasi antarpihak yang terlibat dalam setiap transaksi. S-HTTP terbatas pada *software* khusus yang mengimplementasikannya dan mengenkripsi setiap pesan secara individual. (Dari RSA Cryptography FAQ, hlm. 138).

- S/MIME: *Secure Multipurpose Internet Mail Extension* adalah standar enkripsi yang digunakan untuk mengenkripsi surat elektronik, atau tipe pesan lain di Internet. Ini merupakan standar terbuka yang dikembangkan RSA, sehingga kemungkinan akan kita jumpai di Linux suatu hari. Informasi lebih lanjut tentang S/MIME dapat ditemukan di <http://home.netscape.com/assist/security/smime/overview.html>.

SSL

SSL atau *Secure Sockets Layer* adalah metode enkripsi yang dikembangkan oleh Netscape untuk memberikan keamanan di Internet. Ia mendukung beberapa protokol enkripsi dan memberikan otentikasi client dan server. SSL beroperasi pada layer *transport*, menciptakan saluran enkripsi yang aman untuk data, dan dapat mengenkripsi banyak tipe data. Hal ini dapat dilihat ketika mengunjungi site yang aman untuk melihat dokumen *online* dengan Communicator atau web browser. SSL berfungsi sebagai dasar komunikasi yang aman dengan Communicator, juga dengan enkripsi data Netscape Communication lainnya. Informasi lebih banyak dapat dijumpai di <http://www.consensus.com/security/ssl-talk-faq.html>. Informasi mengenai implementasi keamanan Netscape lainnya dan sebagai titik awal yang baik untuk protokol-protokol ini tersedia di <http://home.netscape.com/info/security-doc.html>.

Kebanyakan otentikasi dan pengiriman data yang dilakukan oleh software server seperti IMAP, POP3, Samba, OpenLDAP, dan Apache, dikirimkan dalam bentuk teks "telanjang" sehingga data tersebut sangat mudah diintip (*sniff*) saat paket data tersebut dikirim lewat jaringan. Banyak

informasi penting yang bersifat rahasia yang dapat dengan mudah di-*sniff* pada saat data tersebut di kirim dari *host* ke *host* yang lain.

Perhatikan contoh potongan *sniffing* terhadap data user dan password dari seseorang yang login ke *host* yang menjalankan web server Apache yang tidak menjalankan SSL:

```
.
43 6F 6E 74 65 6E 74 2D 74 79 70 65 3A
20 61 70 Content-type: ap
70 6C 69 63 61 74 69 6F 6E 2F 78 2D 77
77 77 2D plication/x-www-
66 6F 72 6D 2D 75 72 6C 65 6E 63 6F 64
65 64 0D form-urlencoded.
0A 43 6F 6E 74 65 6E 74 2D 6C 65 6E 67
74 68 3A .Content-length:
20 34 34 0D 0A 44..
75 73 65 72 6E 61 6D 65 3D 61 64 6D 69
6E 26 70 username=admin&p
61 73 73 77 6F 72 64 3D 6D 79 50 61 73
73 30 31 assword=myPass01
26 6C 6F 67 69 6E 3D 4C 6F 67 69 6E
&login=Login
.
```

Untuk menghindari hal ini, digunakanlah teknologi SSL. Data tersebut di enkripsi (diacak) terlebih dahulu sebelum dikirim melalui jaringan, sehingga usaha sniffing seperti contoh di atas hanya akan mendapatkan teks acak yang tidak karuan.

→ Cara kerja SSL

Server mengirim *public key* ke browser. Public key ini digunakan oleh browser untuk melakukan enkripsi terhadap data yang akan dikirim ke server. Data hasil enkripsi tersebut hanya dapat dikembalikan lagi ke data aslinya dengan menggunakan private key yang hanya dimiliki oleh server. Sehingga data yang dienkripsi tersebut hanya dapat di dikembalikan ke bentuk asalnya oleh server saja.

→ Kompilasi, optimasi, dan instalasi openssl

1. Setelah Anda melakukan ekstrak (*un-tar* dan *un-gzipped*) terhadap paket openssl yang Anda *download*, kemudian edit file `tools/c_rehash` dengan mengganti bagian berikut:

```
my $dir = "/usr/local/ssl";
```

menjadi:

```
my $dir = "/usr";
```

karena kita akan menginstalasinya di direktori `/usr`.

2. Sesuaikan lokasi tempat file perl Anda berada, dengan menggunakan util/`perlpath.pl`. Misalnya, bila perl Anda berada pada `/usr/bin`, maka Anda harus memberikan perintah:

```
# perl ./util/perlpath.pl /usr/bin
```

3. OpenSSL harus mengetahui di mana letak kode sumber *library* yang dia butuhkan saat kompilasi. Untuk itu, Anda harus mengetikkan perintah berikut (pada direktori tempat kode sumber ssl Anda berada):

```
# export LD_LIBRARY_PATH='pwd'
```

4. Aturlah konfigurasi `openssl` sesuai dengan sistem Anda. Contoh:

```
# CC="egcs" ./Configure linux-elf -
DSSL_FORBID_ENULL \
-prefix=/usr --openssldir=/etc/ssl
```

Option `-DSSL_FORBID_ENULL` adalah untuk meningkatkan tingkat keamanan kita, dengan menolak null encryption (enkripsi terhadap NULL karakter).

5. Lakukan editing file `Makefile.ssl`, bagian:

```
CC= gcc
```

menjadi

```
CC= egcs
```

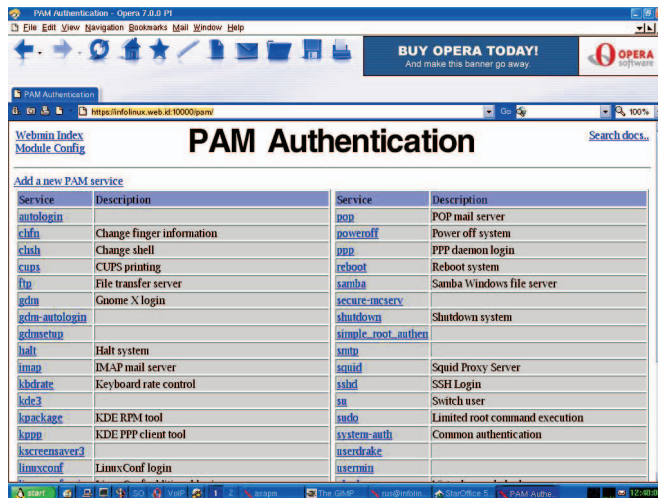
dan untuk CFLAG menjadi:

```
CFLAG= -DTHREADS -D_REENTRANT -
DSSL_FORBID_ENULL -DL_ENDIAN -
DTERMIO -O9-
funroll-loops -ffast-math -malign-double -
mcpu=pentiumpro -march=pentiumpro -
fomit-frame-pointer-
fno-exceptions -Wall -DSHA1_ASM -
DMD5_ASM -DRMD160_ASM
```

juga bagian `PROCESSOR=`

Silakan Anda isi sesuai dengan processor pada mesin Anda. Contoh, bila Anda menggunakan Pentium Pro, isikan baris sebagai berikut:

```
PROCESSOR= 686
```



▲ Konfigurasi PAM dari Webmin

6. Edit file Makefile.ssl bagian MANDIR, dari:

```
MANDIR = $(OPENSSSLDIR)/man
```

menjadi:

```
MANDIR = /usr/man
```

Karena Anda akan menginstal manual pada /usr/man.

7. Lakukan kompilasi dan instalasi:

```
# make -f Makefile
# make test
# make install
# mv /etc/ssl/misc/* /usr/bin/
# rm -rf /etc/ssl/misc/
# rm -rf /etc/ssl/lib/
# rm -f /usr/bin/CA.pl
# rm -f /usr/bin/CA.sh
# install -m 644 libRSAglue.a /usr/lib/
# install -m 644 rsaref/rsaref.h /usr/include/openssl
# strip /usr/bin/openssl
# mkdir -p /etc/ssl/crl
```

Keterangan:

```
# make -f Makefile
```

➔ Perintah ini akan membentuk file library openssl (libcrypto.a, libssl.a) dan file biner openssl. Library akan terbentuk di direktori teratas dari kode sumber openssl, sedangkan file biner berada pada direktori apps.

```
# make test
```

➔ Perintah ini bertujuan utk melakukan pengujian terhadap terhadap library

yang telah dibentuk dengan perintah make -f Makefile.

```
# mv /etc/ssl/misc/* /usr/bin/
```

➔ Perintah ini akan memindah file-file yang ada di direktori /etc/ssl/misc ke dalam direktori: /usr/bin.

```
# rm -rf /etc/ssl/misc/
```

```
# rm -rf /etc/ssl/lib/
```

```
# rm -f /usr/bin/CA.pl
```

```
# rm -f /usr/bin/CA.sh
```

➔ Empat perintah di atas akan menghapus direktori dan file file yang tidak kita perlukan.

```
# install -m 644 libRSAglue.a /usr/lib/
```

```
# install -m 644 rsaref/rsaref.h /usr/include/openssl
```

```
# strip /usr/bin/openssl
```

```
# mkdir -p /etc/ssl/crl
```

➔ Empat perintah terakhir untuk melakukan instalasi lib, header, dan stripping terhadap openssl, dan membuat direktori /etc/ssl/crl.

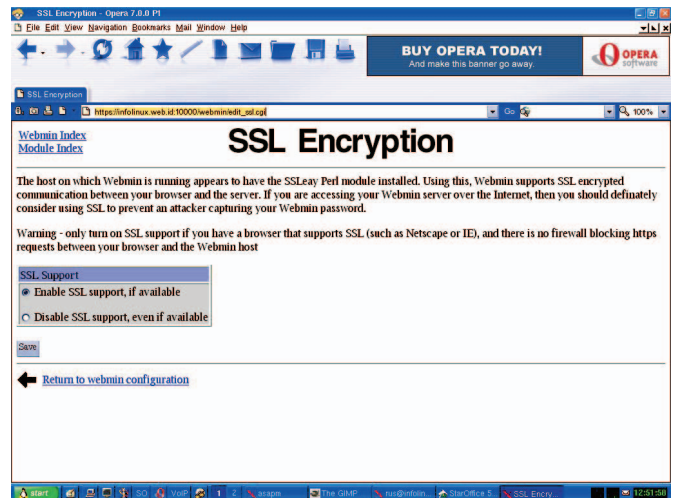
Catatan:

Anda harus menginstal bc-1.05a atau yang lebih baru, karena jika Anda belum menginstalnya, maka selama pengujian library akan muncul pesan kesalahan.

Sekarang masuk ke dalam direktori /etc/ssl dengan mengetikkan perintah:

```
# cd /etc/ssl
```

Lalu lakukan langkah-langkah berikut ini.



▲ SSL untuk Remote Administration

➔ Membuat RSA Private Key

Untuk membuat RSA Private Key, ketikkan perintah berikut dan masukkan key sebanyak dua kali. (Anda berada di direktori /etc/ssl).

```
# openssl genrsa -des3 -out server.key 1024
warning, not much extra random data,
consider using the -rand option
```

Generating RSA private key, 1024 bit long modulus

```
.....++++++
```

```
.....++++++
```

e is 65537 (0x10001)

Enter PEM pass phrase:

Verifying password - Enter PEM pass phrase:

Silakan Anda *back-up* file ca.key dan ingat ingat pass phrase yang Anda masukkan.

➔ Membuat CSR (Certificate Signing Request)

Untuk membuat CSR sesuai dengan RSA Private Key yang telah dibuat sebelumnya, ketikkan perintah:

```
openssl genrsa -des3 -out server.key 1024
```

Kemudian isikan *passphrase*, *data*, dan *challenge password*, contoh:

```
# openssl genrsa -des3 -out server.key 1024
warning, not much extra random data,
consider using the -rand option
```

Generating RSA private key, 1024 bit long modulus

```
.....++++++
```

```
.....++++++
```

```
e is 65537 (0x10001)
Enter PEM pass phrase:
Verifying password - Enter PEM pass phrase:
[root@server ssl]# openssl req -new -key
server.key -out server.csr
Using configuration from /etc/ssl/openssl.cnf
Enter PEM pass phrase:
You are about to be asked to enter informa
tion that will be incorporated
into your certificate request.
What you are about to enter is what is called
a Distinguished Name or a DN.
There are quite a few fields but you can
leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
--
Country Name (2 letter code) [AU]:ID
State or Province Name (full name) [Some-
State]:Jawa Tengah
Locality Name (eg, city) []:Semarang
Organization Name (eg, company) [Internet
Widgits Pty Ltd]:Atlantis
Organizational Unit Name (eg, section) []:IT
Common Name (eg, YOUR name) []:Rootman
Email Address []:rootman@white-star.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:password
An optional company name []:Kompeni
```

➔ Membuat sertifikat Self-Signed

Untuk membuat Sertifikat Self-Signed dengan RSA key dari CA, ketikkan perintah:

```
# openssl req -new -x509 -days 365 -key
ca.key -out ca.crt
```

Kemudian isilah *pass phrase* dan data sertifikat seperti pada saat Anda membuat CSR (*Certificate Signing Request*).

Setelah Key dan sertifikat terbentuk, pindahkan masing-masing ke dalam direktori tempatnya, contoh:

```
# mv server.key private/
# mv ca.key private/
# mv ca.crt certs/
```

➔ Membuat server.crt untuk Apache

Untuk menerapkan openSSL pada web server Apache, Anda membutuhkan server.crt. Untuk membuatnya, ikuti langkah di bawah ini:

Buat skrip bernama sign.sh, yang isinya sebagai berikut:

```
----- begin of sign.sh -----
#!/bin/sh
# make sure environment exists
if [ ! -d ca.db.certs ]; then
    mkdir ca.db.certs
fi
if [ ! -f ca.db.serial ]; then
    echo '01' > ca.db.serial
fi
if [ ! -f ca.db.index ]; then
    cp /dev/null ca.db.index
fi
# create an own SSLeay config
cat > ca.config <
```

Kemudian jalankan:

```
# sign.sh
Using configuration from ca.config
Enter PEM pass phrase:
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName       :PRINTABLE:'ID'
stateOrProvinceName :PRINTABLE:'Jawa Tengah'
localityName       :PRINTABLE:'Semarang'
organizationName   :PRINTABLE:'Atlantis'
organizationalUnitName:PRINTABLE:'IT'
commonName         :PRINTABLE:'Rootman'
emailAddress        :IA5STRING:'rootman@white-star.com'
Certificate is to be certified until Dec 7
09:35:28 2002 GMT (365 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified,
commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
server.crt: OK
```

Maka akan terbentuk file bernama server.crt, lalu pindahkan sertifikat ini (server.crt), ke direktori */etc/ssl/certs*.

```
# mv /etc/ssl/server.crt /etc/ssl/certs
```

Kemudian hapus file server.csr dan ubah permisi dari 4 file SSL (key dan crt) menjadi 600:

```
# rm /etc/ssl/server.csr
# chmod 600 /etc/ssl/certs/ca.crt
```

```
# chmod 600 /etc/ssl/certs/server.crt
# chmod 600 /etc/ssl/private/ca.key
# chmod 600 /etc/ssl/private/server.key
```

Sampai dengan langkah ini, file sertifikat dan key untuk SSL terbentuk semua. Sebagai contoh, jika Anda akan menerapkan ssl pada Apache Anda, silakan tambahkan baris dalam *httpd.conf* sebagai berikut.

```
SSLCertificateFile /etc/ssl/certs/server.crt
SSLCertificateKeyFile /etc/ssl/private/
server.key
```

Setelah Anda melakukan *restart* Apache, maka web server apache Anda sudah siap untuk melakukan *secure transaction*, dengan enkripsi openSSL 128 bit.

Mendeteksi gangguan secara dini

Penting bagi administrator sistem untuk memeriksa sistem sedini mungkin dari tindakan penyusupan yang mungkin saja dilakukan oleh user lokal maupun penyusup dari luar. Berikut ini prosedur pendeteksian gangguan secara dini:

1. File-file *read-writable*, utamanya file sistem, dapat menjadi lubang keamanan jika seorang cracker memperoleh akses ke sistem Anda dan memodifikasinya. Selain itu, direktori *read-writable* berbahaya karena memungkinkan cracker menambah atau menghapus file sesuai keinginannya. Untuk mencari seluruh file *world-writable* di sistem Anda, gunakan perintah berikut:

```
[root@localhost: ~]# find / -perm -2 -
print
```

2. File-file SUID dan SGID pada sistem Anda merupakan potensi risiko keamanan, dan harus diawasi dengan baik. Oleh karena program-program ini memberi izin khusus bagi pemakai yang mengeksekusinya, maka perlu dipastikan bahwa program yang tidak aman tidak diinstalasi. Trik favorit cracker adalah mengeksploitasi program SUID "root", lalu meninggalkan program SUID sebagai *backdoor* untuk masuk di saat lain,

meski lubang yang asli telah ditutup. Carilah seluruh program SUID/SGID di sistem Anda, dan catatlah, sehingga Anda mengerti setiap perubahan yang dapat mengindikasikan penyusup potensial. Gunakan perintah berikut untuk mencari seluruh program SUID/SGID di sistem Anda:

```
[root@localhost: ~] # find / -type f \( -perm -04000 -o -perm -02000 \)
```

3. File-file yang tidak ada pemiliknya juga dapat menjadi indikasi penyusup telah mengakses sistem Anda. Anda dapat menemukan file-file di sistem Anda yang tidak memiliki pemilik, atau milik suatu kelompok dengan perintah:

```
[root@localhost: ~] # find / -nouser -o -nogroup -print
```

4. Mencari file .rhosts seharusnya menjadi bagian tugas reguler Anda sebagai administrator sistem, karena file ini tidak diizinkan ada di sistem Anda. Ingat, cracker hanya perlu satu

rekening tidak aman untuk secara potensial memperoleh akses ke seluruh jaringan Anda. Anda dapat melihat seluruh file .rhosts di sistem Anda dengan perintah:

```
[root@localhost: ~] # find /home -name .rhosts -print
```

Penggunaan Tripwire

Cara baik lain untuk mendeteksi serangan lokal (dan juga jaringan) pada sistem Anda adalah dengan menjalankan pemeriksa integritas seperti *Tripwire*. Tripwire menjalankan sejumlah *checksum* di seluruh file *biner* dan *config* penting Anda dan membandingkannya dengan database terdahulu, yang diketahui baik sebagai referensi. Oleh karena itu, setiap perubahan dalam file akan diketahui.

Untuk amannya, Anda dapat menginstalasi Tripwire ke floppy, dan kemudian mengeset *write protect* secara fisik pada floppy. Dengan demikian, penyusup tidak dapat mengganggu Tripwire atau mengubah database. Sekali

Anda telah memiliki setup Tripwire, merupakan ide yang baik untuk menjalankannya sebagai tugas administrasi keamanan normal Anda, untuk melihat jika ada perubahan.

Anda bahkan dapat menambahkan *entry crontab* untuk menjalankan Tripwire dari floppy setiap malam dan mengirimkan hasilnya kepada Anda di pagi hari.

Contoh:

```
[root@localhost: ~] # crontab -e
MAILTO=masaji
* 02 * * * root /usr/local/adm/tcheck/tripwire
```

Perintah di atas akan mengakibatkan sistem memberi laporan kepada user masaji setiap hari jam dua pagi.

Tripwire dapat pula menjadi petunjak yang baik untuk mendeteksi penyusup sebelum Anda mengetahuinya. Oleh karena banyaknya file yang berubah pada rata-rata sistem, Anda harus berhati-hati tentang aktivitas cracker dan apa yang Anda lakukan. 🐧

R. Kresno Aji (masaji@ai.co.id)

LINUX PROFESSIONAL

Lembaga Pendidikan Komputer Nurul Fikri

www.nurulfikri.com

info@nurulfikri.com

PROGRAM PROFESI 200 JAM

Linux Server Development (LSD)

Materi:

- ⇒ Linux Fundamental (Linux Basic, X Window, Linux SysAdmin, Networking)
- ⇒ Internet
- ⇒ Shell Programming
- ⇒ Advanced System Administration
- ⇒ Security
- ⇒ Advanced Networking
- ⇒ HTML, CSS & Javascript
- ⇒ PHP & MySQL
- (Linux Complete, Security, Web & Database)

LAYANAN MIGRASI KE LINUX

Regular & In House Training

- ⇒ Server (PDC, Database, File, Mail, Proxy, Web)
- ⇒ Security & Networking
- ⇒ Toko-toko komputer untuk Instalasi Linux

Segera dibuka di Jakarta dan Cilegon Program Profesi 1 Tahun Komputer dan Informatika Berbasis Linux plus Windows

Hubungi:

- Jl. Margonda Raya No. 522 - UI Depok, Telp. (021) 7874223/4 Fax. 7874224
- Jl. Mampang Prapatan X/4 Jakarta Selatan, Telp. (021) 7975235, 7947115
- Jl. Beringin Raya 14a, Perumnas I Tangerang, Telp. (021) 5589884
- Graha Sucofindo, Jl. Jend. Ahmad Yani No. 106 Cilegon - Banten, Telp/Fax. 0254-374345, 374456

