BAB IV

IMPLEMENTASI PERANGKAT AUDIT NETWORK SECURITY

Pada bab keempat ini, akan membahas hal-hal yang berkaitan dengan implementasi PANS yang meliputi deskripsi lingkungan implementasi dan implementasi rancangan itu sendiri.

4.1 Deskripsi Lingkungan Implementasi

Deskripsi lingkungan ini adalah deskripsi lingkungan yang digunakan untuk mengimplementasi hasil rancangan dari Bab III. Sesuai dengan jenisnya lingkungan implementasi dilihat dari dua sisi, yaitu sisi perangkat keras dan sisi perangkat lunak.

4.1.1 Lingkungan Perangkat Keras

Perangkat Audit *Network security* yang telah penulis rancang pada Bab III, diimplementasikan dalam sebuah lingkungan perangkat keras berspesifikasi sebagai berikut :

1. PC dengan *processor* Intel Pentium IV 1,8 GHz
2. RAM 256 MB
3. Harddisk kapasitas 80 GB untuk data dan sistem, 512 MB disediakan untuk *swap*.
4. Monitor SVGA
5. *Keyboard* dan *mouse* standar

4.1.2 Lingkungan Perangkat Lunak

Perangkat lunak yang dipasang penulis pada sistem komputer yang digunakan untuk membangun server PANS adalah sebagai berikut :

1. Sistem Operasi FreeBSD 4.6-STABLE
2. Nessus-1.2.5, sebuah *tool* yang berfungsi untuk memeriksa aspek keamanan sebuah komputer.
3. PHP-4.2.3, sebuah bahasa pemrograman CGI (*Common Gateway Interface*) yang digunakan sebagai penghubung antara *web server* Apache dengan *database* MySQL dan membuat halaman *web* menjadi dinamis.
4. MySQL-3.23.51, sebuah aplikasi *database* untuk Unix.
5. Apache-1.3.26, sebuah *web server* untuk Unix.

4.2 Implementasi Rancangan

Sesuai gambar 3.1, implementasi terbagi menjadi tiga bagian, yaitu instalasi aplikasi pendukung (Nessus, Apache, PHP, MySQL, sistem cron), implementasi modul PANS, dan implementasi sistem *database* PANS.

4.2.1 Instalasi Aplikasi Pendukung

Beberapa aplikasi yang mendukung PANS yang perlu di-*install* adalah MySQL, Apache, PHP, dan Nessus. Selain itu diperlukan pula setting konfigurasi untuk sistem cron Unix yang sudah ada. Sebelum aplikasi pendukung dipasang, pada *server* FreeBSD diperlukan *user* baru yang khusus diperlukan untuk PANS. Oleh karena itu dibuat *user* dengan perintah :

```
# pw groupadd www
# pw useradd -d /home/www -g www www
# passwd www
# chmod 711 /home/www
```

1. Instalasi MySQL-3.23.51

   Aplikasi *database* MySQL tersedia dalam bentuk paket tarball (MySQL-3.23.51.tar.gz). Urutan instalasi MySQL adalah sebagai berikut :

```
# pw groupadd mysql
# pw useradd -g mysql mysql
# tar -xvzf MySQL-3.23.51.tar.gz
# ./configure --prefix=/usr/local/mysql
# make && make install
# scripts/mysql_install_db
# chown -R root /usr/local/mysql
# chown -R mysql /usr/local/mysql/var
# chgrp -R mysql /usr/local/mysql
# echo "/usr/local/mysql/bin/safe_mysqld --user=mysql &" >
  /usr/local/etc/rc.d/mysql.sh
```

   Selanjutnya MySQL yang sudah ter-*install* memerlukan konfigurasi lebih lanjut supaya bisa digunakan oleh PANS. Langkah-langkah berikut dilakukan :

```
Mengubah password root MySQL :
   # mysqladmin -u root -p password 'rootmysql'

Membuat database baru untuk PANS :
   # mysql -u root -p
   mysql> create database db_pans;
                          a

Menambah user untuk PANS pada MySQL :
   mysql> grant all privileges on db_pans.* to pans@localhost identified by 'pansjuga';
                                 a          b                                  c
```

   Bagian a adalah nama *database* untuk PANS, b adalah nama *user* untuk PANS, dan c adalah *password* untuk *user* PANS. Ketiga variabel tersebut bisa diubah, tetapi harus sama seperti yang ada pada file config.php.

2. Instalasi Apache-1.3.26

   Apache *web server* untuk FreeBSD tersedia dalam bentuk tarball (apache-1.3.26.tar.gz). Langkah-langkah instalasinya adalah sebagai berikut :

```
# tar -xvzf apache-1.3.26.tar.gz
# cd apache-1.3.26/
# ./configure --prefix=/usr/local/apache --enable-module=so
# make
# make install
# echo "/usr/local/apache/bin/apachectl start"> /usr/local/etc/rc.d/apache.sh
```

3. Instalasi PHP-4.2.3

PHP juga tersedia dalam bentuk tarball (php-4.2.3.tar.gz). Untuk instalasi PHP terbagi menjadi tiga tahap, yaitu tahap instalasi PHP sebagai modul dari Apache, tahap instalasi PHP sebagai command line interpreter sehingga bisa dijalankan dari lingkungan *shell* Unix, dan tahap konfigurasi *web server* Apache sehingga bisa menjalankan PHP. Lebih lengkapnya adalah sebagai berikut :

```
Tahap I - Instalasi PHP sbg Modul Apache
   # tar -xvzf php-4.2.3.tar.gz
   # cd php-4.2.3/
   # ./configure --with-apxs=/usr/local/apache/bin/apxs –with-mysql
   # make && make install

Tahap II - Instalasi PHP sbg CLI
   # make clean
   # ./configure --with-mysql
   # make && make install

Tahap III - Konfigurasi Apache untuk menjalankan PHP
Mengubah file /usr/local/apache/conf/httpd.conf dengan menggunakan teks editor
standar Unix yaitu vi. Baris-baris yang bersesuain diubah nilainya menjadi sbb :
   User www
   group www
   DocumentRoot "/home/www"
   DirectoryIndex index.php
Kemudian file tsb disimpan dan pada shell Unix dijalankan perintah berikut :
   # echo "AddType application/x-httpd-php .php" >> /usr/local/apache/conf/hhttpd.conf
   # /usr/local/apache/bin/apachectl restart
```

4. Instalasi Nessus-1.2.5

Aplikasi Nessus disebarluaskan dalam dua macam bentuk paket instalasi, yaitu dalam bentuk tarball (*.tar.gz) yang terbagi menjadi empat file tarball (nessus-libaries-1.2.5.tar.gz, libnasl-1.2.5.tar.gz, nessus-core.1.2.5.tar.gz, dan nessus-plugins.1.2.5.tar.gz) dan dalam bentuk satu file paket instalasi bernama nessus-

installer.sh. Supaya lebih mudah digunakan paket instalasi yang kedua. Perintah berikut dijalankan dalam *shell* FreeBSD (tanda # menunjukkan *prompt-shell* Unix) :

```
# sh nessus-installer.sh
```

Aplikasi Nessus (nessusd) harus diaktifkan dahulu di dalam sistem FreeBSD dalam bentuk *daemon* sebelum bisa digunakan oleh PANS. Oleh karena itu file nessusd (singkatan dari Nessus *Daemon*) sebaiknya dijalankan saat proses *booting server*, dengan menjalankan perintah berikut :

```
# echo "/usr/local/sbin/nessusd -D "> /usr/local/etc/rc.d/nessus.sh
```

Selanjutnya Nessus juga memerlukan *user* khusus sebelum bisa digunakan PANS. Oleh karena itu kita membuat *user* Nessus dengan perintah berikut :

```
# /usr/local/sbin/nessus-adduser
Login : nessuspans
Password : nessuspansjuga
```

User dan *password* di atas adalah contoh, bisa diubah dengan yang lain, tetapi yang perlu diperhatikan adalah *user* dan *password* yang digunakan disini (*server* PANS) harus sama dengan yang digunakan pada *client* PANS dan pada file config.php.

5. Konfigurasi sistem cron

Secara default sistem cron sudah terpasang pada sistem operasi FreeBSD. Untuk keperluan PANS dilakukan konfigurasi yang memungkinkan PANS melakukan dua hal penting yaitu kemampuan *scheduled scan* dan proses *update plugins*. Kedua hal tersebut sudah dibahas dengan jelas pada Bab III yaitu pada modul 3.1.2 dan 7.1.

4.2.2 Implementasi Modul PANS

Berikut adalah implementasi modul-modul PANS yang ada pada Bab III, dalam bentuk *listing* program PHP. Semua program PHP ini berbentuk file dan disimpan dalam direktori /home/www :

4.2.2.1 Modul Manajemen *User*

Modul Manajemen *User* mempunyai lima unit fungsional yaitu autentifikasi *user, register user*, hapus *user*, *list user login*, dan ubah *password* :

1.1 Autentifikasi *User*

Sub-unit fungsi autentifikasi *user* adalah :

1.1.1 *Login*

Nama File	login.php
Deskripsi	*Form* pengisian *user name* dan *password* untuk masuk ke dalam sistem PANS.
Input	*User name* dan *password*.
Output	*Input* untuk fungsi validasi lebih lanjut pada file login_check.php.
Algoritma	Lampiran A.1.

1.1.2 *Login* Check

Nama File	login_check.php
Deskripsi	Fungsi untuk validasi proses *login*.
Input	*User name* dan *password* dari file login.php.
Output	Validasi benar/salah, bila benar maka sistem sekaligus akan membuat *session* ID awal untuk *user* yang berhasil *login*.
Algoritma	Lampiran A.2.

1.1.3 *Logout*

Nama File	logout.php
Deskripsi	Fungsi untuk keluar dari sistem PANS.
Input	*Session* ID yang masih valid dan IP *Address browser* yang menggunakan *session* ID tersebut.
Output	*Session* ID yang dimaksud terhapus dari *database* dan *user* ter-*logout*.
Algoritma	Lampiran A.3.

1.2 *Register User*

Nama File	user_register.php
Deskripsi	Menambah *user* PANS (khusus untuk *administrator* PANS).
Input	*user name*, *password*, *email address*, dan *real name.*
Output	Memasukkan data *input* ke dalam *database*.
Algoritma	Lampiran A.4.

1.3 Hapus *User*

Nama File	delete_user.php
Deskripsi	Menghapus *user* PANS yang sudah terdaftar (khusus untuk *administrator* PANS).
Input	*User name*, *password*, *email address*, dan *real name.*
Output	Data *user* yang dihapus hilang dari *database*.
Algoritma	Lampiran A.5.

1.4 *List User Login*

Nama File	list_user.php
Deskripsi	Melihat semua *user login* PANS yang terdaftar, melihat *user* yang sedang *login.*
Input	-
Output	Tampilan pada *web*.
Algoritma	Lampiran A.6.

1.5 Ubah *Password*

Nama File	change_password.php
Deskripsi	Mengubah *password user* PANS.
Input	*Password* lama dan *password* baru.
Output	*Password* baru tersimpan ke dalam *database*.
Algoritma	Lampiran A.7.

4.2.2.2 Modul *Scan*

Modul *Scan* memiliki lima unit fungsional yaitu penentuan *target*, penentuan *client*, konfigurasi *scan*, proses *scan*, dan hasil *scan* ke *database* :

8.1 Penentuan *Target*

Nama File	instant_scan_login.php (instant_scan_nologin.php)
Deskripsi	*Form* pengisian IP *address target* yang akan diperiksa dan pilihan *level scan*.
Input	IP *address* dan *level scan*.
Output	*Input* untuk instant_scan_login(nologin)_check.php, berupa IP *address*, *client*, dan *level scan*.
Algoritma	Lampiran A.8.

8.2 Penentuan *Client*

Nama File	client_insscan.php
Deskripsi	Menentukan *client* untuk *instant scan*.
Input	IP *address client* yang sudah terdaftar dalam *database*.
Output	Satu *client* diset sebagai *client* dan menyimpannya dalam *database* tersendiri.
Algoritma	Lampiran A.9.

8.3 Konfigurasi *Scan*

Nama File	list_plugins.php (untuk *scan login*) list_plugins_nl.php (untuk *scan no login*)
Deskripsi	Mengatur *level scan* dengan cara memilih *level safe*, *danger*, atau memilih per *plugin* yang akan digunakan.
Input	-
Output	Sebuah file konfigurasi pada direktori scan_option/ yaitu nessus_login.rc (untuk *scan login*) atau nessus_nologin.rc (untuk *scan nologin*).
Algoritma	Lampiran A.10 dan Lampiran A.11.

8.4 Proses *Scan*

Proses *scan* terdiri dari tiga macam yaitu *scan login*, *scan nologin*, dan *scheduled scan*.

Nama File	instant_scan_login_check.php
Deskripsi	Proses *scan* secara langsung.
Input	*Output* dari instant_scan_login.php.
Output	Aplikasi Nessus aktif melakukan *scan* dan membuat file hasil *scan* berformat NSR.
Algoritma	Lampiran A.12.

Nama File	instant_scan_nologin_check.php
Deskripsi	Proses *scan* secara langsung.
Input	*Output* dari instant_scan_nologin.php.
Output	Aplikasi Nessus aktif melakukan *scan* dan membuat file hasil *scan* berformat NSR.
Algoritma	Lampiran A.13.

Nama File	cron_for_(ipaddress)
Deskripsi	Proses *scan* secara terjadwal. Berbeda dengan jenis *scan* berikutnya, *scan* ini dijalankan dari sebuah file cron kecil yang berisi perintah php yang diekseskusi bukan oleh *web server*, tetapi oleh Unix-*shell*. Pada algoritma di bawah dimisalkan *host* yang akan di*scan* adalah 167.205.49.144, sedangkan *client* nya adalah 167.205.49.142.
Input	*Output* dari file add_schedule_host.php.
Output	Aplikasi Nessus aktif melakukan *scan* dan membuat file hasil *scan* berformat NSR.
Algoritma	Lampiran A.14.

8.5 Hasil *Scan* ke *Database*

Untuk memasukkan hasil *scan* ke *database* terbagi menjadi tiga macam fungsi, hal ini disebabkan terdapat tiga macam *scan* yang ada dalam PANS, yaitu *scan login*, *scan nologin*, dan *scheduled scan*. Dimana ketiga macam *scan* tersebut melibatkan tabel *database* yang berbeda.

Nama File	login_insscan_result_to_db.php
Deskripsi	Memasukkan hasil *instant scan login* ke dalam *database* (tabel scan_login_header dan scan_login_detail).
Input	Hasil *scan* Nessus yang berupa file berformat NSR.
Output	Data di dalam *database*.
Algoritma	Lampiran A.15.

Nama File	instant_scan_result_to_db.php
Deskripsi	Memasukkan hasil *instant scan nologin* ke dalam *database* (tabel scan_nologin_header dan scan_nologin_detail).
Input	Hasil *scan* Nessus yang berupa file berformat NSR.
Output	Data di dalam *database*.
Algoritma	Lampiran A.16.

Nama File	schedule_scan_result_to_db.php
Deskripsi	Memasukkan hasil *scheduled scan* ke dalam *database* (tabel report_header dan report_detail).
Input	Hasil *scan* Nessus yang berupa file berformat NSR.
Output	Data di dalam *database*.
Algoritma	Lampiran A.17.

4.2.2.3 Modul *Scheduling Scan*

Modul *Scheduling Scan* memiliki tiga unit fungsional yaitu atur *schedule*, lihat *schedule*, dan hapus *schedule* :

3.1 Atur *Schedule*

Nama File	add_schedule_host.php
Deskripsi	Untuk melakukan pengaturan jadwal *scan*.
Input	Pilihan waktu *schedule*.
Output	File cron (cron_for_(ipaddress)) pada direktori *web*.
Algoritma	Lampiran A.18.

3.2 Lihat *Schedule*

Nama File	lihat_schedule_host.php
Deskripsi	Melihat semua *host* yang sudah ditentukan untuk diperiksa secara rutin.
Input	-
Output	Tampilan pada *web*.
Algoritma	Lampiran A.19.

3.3 Hapus *Schedule*

Nama File	delete_schedule_host.php
Deskripsi	Menghapus jadwal *scan* rutin pada sebuah *host*.
Input	Jadwal *scan* sebuah *host* yang terdaftar dalam *database*.
Output	Jadwal *scan* hilang dari *database* dan file cron_for_(ipaddress) untuk *host* yang bersangkutan terhapus.
Algoritma	Lampiran A.20.

4.2.2.4 Modul *Report*

Modul *Report* memiliki empat unit fungsional yaitu : ambil dari *database*, bandingkan dengan ICAT, tampilkan pada *web*, dan hapus *report*. Unit fungsional 4.1, 4.2, dan 4.3 menyatu dalam sebuah file, sedangkan unit 4.4 berada dalam file terpisah.

4.1-4.3 Ambil dr *Database*, Bandingkan dgn ICAT, Tampilkan pada *Web*

Nama File	report_host.php
Deskripsi	Menampilkan semua hasil *scan* yang sudah pernah dilakukan.
Input	IP *address target host*, IP *address* yang melakukan *scan*, waktu *scan*, dan jenis *scan* (misalnya *scan login*, tanpa *login*, atau *scan* terjadwal).
Output	Tampilan pada *web*.
Algoritma	Lampiran A.21.

4.4 Hapus *Report*

Nama File	login_del_report.php
Deskripsi	Menghapus laporan hasil *scan*.
Input	IP *address target host*, IP *address* komputer yang melakukan *scan*, waktu *scan*, dan jenis *scan*.
Output	Laporan *scan* yang dimaksud terhapus dari *database*.
Algoritma	Lampiran A.22.

4.2.2.5 Modul Referensi ICAT

Modul Referensi ICAT memiliki empat unit fungsional yaitu : tambah *vulnerability,* edit *vulnerability,* hapus *vulnerability,* dan cari *vulnerability* :

8.1 Tambah *Vulnerability*

Nama File	insert_vuln.php
Deskripsi	Menambah data *vulnerability.*
Input	CVE ID, *Publish Date,* CVE *Description, Severity, Link* 1, *Link* 2, *Vulnerable* OS, *Specific Component*.
Output	Penambahan data dalam *database*.
Algoritma	Lampiran A.23.

8.2 Edit *Vulnerability*

Nama File	edit_vuln.php
Deskripsi	Mengubah sebuah data *vulnerability.*
Input	CVE ID, *Publish Date,* CVE *Description, Severity, Link* 1, *Link* 2, *Vulnerable* OS, *Specific Component*.
Output	Perubahan data disimpan dalam *database*.
Algoritma	Lampiran A.24.

8.3 Hapus *Vulnerability*

Nama File	delete_vuln_confirm.php
Deskripsi	Menghapus sebuah data *vulnerability.*
Input	CVE ID.
Output	Data *vulnerability* yang dimaksud terhapus dari *database*.
Algoritma	Lampiran A.25.

8.4 Cari *Vulnerability*

Nama File	search_cve.php
Deskripsi	Mencari data *vulnerability.*
Input	Kata kunci pencarian.
Output	Tampilan pada *web* semua data *vulnerability* yang cocok dengan kata kunci pencarian.
Algoritma	Lampiran A.26.

4.2.2.6 Modul Manajemen *ClientTarget*

Modul Manajemen *ClientTarget* memiliki enam unit fungsional yaitu view *client*/*target*, tambah *client*/*target*, ubah *client*/*target*, hapus *client*/*target*, restrict *target*, dan tentukan default *client* :

8.1 View *Client*/*Target*

Nama File	browse_client.php
Deskripsi	Melihat semua *client* yang sudah terdaftar dalam *database*.
Input	-
Output	Tampilan pada *web*.
Algoritma	Lampiran A.27.

Nama File	browse_host.php
Deskripsi	Melihat semua *target host* yang sudah terdaftar dalam *database*
Input	-
Output	Tampilan pada *web*.
Algoritma	Lampiran A.28.

8.2 Tambah *Client*/*Target*

Nama File	add_client.php
Deskripsi	Menambah sebuah *client*.
Input	IP *Address client*, *Client hostname*, *Comment*.
Output	Data tersebut masuk ke dalam *database*.
Algoritma	Lampiran A.29.

Nama File	add_host.php
Deskripsi	Menambah sebuah *target host*.
Input	IP *Address target host*, *Client* yang dipakai, Status restriksi.
Output	Data tersebut masuk ke dalam *database*.
Algoritma	Lampiran A.30.

8.3 Ubah *Client*/*Target*

Nama File	edit_client.php
Deskripsi	Mengubah data sebuah *client*.
Input	IP *Address client*.
Output	Data *client* yang dimaksud berubah.
Algoritma	Lampiran A.31.

Nama File	edit_host.php
Deskripsi	Mengubah data sebuah *target host*.
Input	IP *Address target host*.
Output	Data *target host* yang dimaksud berubah.
Algoritma	Lampiran A.32.

8.4 Hapus *Client*/*Target*

Nama File	del_client.php
Deskripsi	Menghapus data sebuah *client* dari *database*.
Input	IP *Address client*.
Output	Data *client* yang dimaksud terhapus dari *database*.
Algoritma	Lampiran A.33.

Nama File	del_host.php
Deskripsi	Menghapus data sebuah *target host* dari *database*.
Input	IP *Address target host*.
Output	Data *target host* yang dimaksud terhapus dari *database*.
Algoritma	Lampiran A.34.

8.5 Restrict *Target*

Bagian ini sudah terintegrasi pada unit fungsional 6.2 (Tambah *Client*/*Target*) dan 6.3 (Ubah *Client*/*Target*).

8.6 Tentukan Default *Client*

Nama File	client_insscan.php
Deskripsi	Menentukan *client* yang dipakai untuk *instant scan*.
Input	IP *Address client*.
Output	Default *client* berubah.
Algoritma	Lampiran A.35.

4.2.2.7 Modul *Update*

Modul *Update* memiliki dua unit fungsional yaitu *download plugins* dan masukkan *database* :

8.1 *Download Plugins*

Bagian ini memanfaatkan fungsi *built-in* milik Nessus atau *download* secara manual dari site Nessus.

8.2 Masukkan *Database*

Nama File	get_plugins.php
Deskripsi	File induk yang berfungsi memasukkan semua *plugins* yang berada pada dir /usr/local/lib/nessus/plugins/ ke dalam *database*, sekaligus membaca data pada masing-masing *plugins*.
Input	-
Output	Data detail *plugins* masuk ke dalam *database*.
Algoritma	Lampiran A.36.

Nama File	parse_plugins.php
Deskripsi	Membaca secara detail data yang disertakan dalam setiap file *plugin*.
Input	File *plugin*.
Output	Data detail *plugins* masuk ke dalam *database*.
Algoritma	Lampiran A.37.

Nama File	find_danger_plugins.php
Deskripsi	Menentukan tingkat *scan* yang mampu dilakukan oleh masing-masing *plugin*.
Input	File *plugin*.
Output	Data detail *plugins* masuk ke dalam *database*.
Algoritma	Lampiran A.38.

4.2.2.8 Modul Lain-lain (Tambahan)

Modul Tambahan ini memiliki lima unit fungsional yaitu : hapus proses *error*, tampilkan file sistem, hapus file sistem, pemakaian *session* ID, dan *auto logout* :

8.1 Hapus Proses *Error*

Nama File	process_admin.php
Deskripsi	Melihat proses *scan* yang bermasalah.
Input	-
Output	Tampilan pada *web*.
Algoritma	Lampiran A.39.

Nama File	del_process.php
Deskripsi	Menghapus proses *scan* yang bermasalah.
Input	IP *Address* komputer yang melakukan *scan*, IP *Address target host*, Waktu mulai *scan*, Waktu akhir *scan*, Jenis *scan* yang dilakukan.
Output	Proses *scan* yang bermasalah terhapus dari *database*.
Algoritma	Lampiran A.40.

8.2 Tampilkan File Sistem

Nama File	view_file.php
Deskripsi	Membaca isi file sistem.
Input	Nama file.
Output	Tampilan pada *web*.
Algoritma	Lampiran A.41.

8.3 Hapus File Sistem

Nama File	List_file.php
Deskripsi	Melihat dan menghapus file sistem.
Input	Nama file.
Output	File terhapus dari direktori scan_option/.
Algoritma	Lampiran A.42.

8.4 Pemakaian *Session* ID

Nama File	login_check.php
Deskripsi	Pembuatan awal dan pendaftaran *session*.
Input	Waktu *login* pertama kali.
Output	*Session* terbuat dan disimpan di dalam *database*.
Algoritma	Lampiran A.43.

Nama File	key.php
Deskripsi	Pemeriksaan terhadap *Session* ID sebagai autentifikasi pada file-file khusus untuk *user login*.
Input	*Session* ID.
Output	File khusus bisa diakses.
Algoritma	Lampiran A.44.

8.5 *Auto Logout*

Nama File	config.php
Deskripsi	Setting waktu *idle* untuk proses *auto logout*.
Input	-
Output	Bila waktu *idle* sudah tercapai maka *user* harus *login* kembali.
Algoritma	Lampiran A.45.

4.2.3 Implementasi Sistem *Database* PANS

Aplikasi PANS ini menggunakan *database* MySQL. Untuk mengimplementasi sistem PANS diperlukan sejumlah tabel dalam database tersebut. Berikut adalah tabel-tabel tersebut disertai dengan penjelasan fungsi PANS pada Bab III yang menggunakannya, disusun menurut abjad :

1. Tabel client_insscan

   Untuk tabel ini yang diperlukan sistem adalah id dan *client*, fungsinya adalah untuk menyimpan *client* default hasil dari fungsi 6.6 (Tentukan Default *Client*). Isi dari tabel ini selalu dipertahankan satu *record*, karena fungsinya untuk menyimpan satu *client* yang telah ditentukan. *Client* ini digunakan dalam proses *scan* untuk *user nologin*. Berikut adalah contoh tabelnya :

id	*client*
1	167.205.49.142

2. Tabel client_list

   Tabel berisi daftar *client* yang dimasukkan dalam sistem PANS ini memerlukan *field* IP, *hostname*, dan *comment*. Disini *field* IP diset sebagai *primary key* dan bersifat unik (*unique key*), jadi dalam satu tabel ini tidak boleh ada *record* dengan IP yang sama. Fungsi yang menggunakan tabel ini adalah fungsi 6.1-6.4, dan 6.6. Contoh tabel client_list :

ip	*hostname*	*comment*
167.205.49.146	kuda.ee.itb.ac.id	*Server* sekaligus *client* PANS
167.205.49.142	gajah.ee.itb.ac.id	*Client* 1

3. Tabel host_list

   Hampir sama dengan tabel client_list tetapi pada tabel ini berisi komputer-komputer *target* (*host*) yang siap untuk diaudit secara berkelanjutan oleh PANS. *Field* yang dibutuhkan adalah *client*, *target*, dan restricted. *Field target* diset sebagai *primary key* dan *unique key*. Sedangkan *field restricted* memiliki tipe data enum yang berisi Y atau N, bila sebuah *host* memiliki status restriksi Y(es) berarti tidak bisa di*scan* oleh *user nologin* dan bila statusnya N(o) berarti komputer *target* ini bebas untuk di*scan* oleh siapa saja. Fungsi yang memakai tabel ini adalah fungsi 6.1-6.6. Contoh tabelnya adalah sebagai berikut :

client	*target*	restricted
167.205.49.142	167.205.49.144	N
167.205.49.142	167.205.49.147	Y

4. Tabel *plugins*

Tabel ini menyimpan semua *plugin* Nessus yang siap digunakan oleh sistem PANS. *Field* yang terdapat di dalamnya adalah cat, spec1, note, owner, spec2, rev, cve, detail, dan *safe. Field-field* ini berisi data yang dibawa oleh setiap *plugin* Nessus, tidak semua *field* dipakai dalam sistem, hanya *field* spec1, spec2, dan *safe.* Fungsi yang menggunakannya adalah 2.3 dan 7.2. Contoh tabel *plugins* :

spec1	spec2	*safe*
BIND vulnerable	Checks the remote BIND version	*safe*
BlackIce DoS (ping flood)	Ping flood the remote machine and kills BlackIce	*danger*

5. Tabel report_detail

Tabel report_detail digunakan untuk menyimpan data detail hasil *scheduled scan*. Terdiri dari *field-field* : report_id, *host*, *port*, *type*, data, *solution*, CVE_ID, risk_factor. Fungsi yang menggunakannya adalah fungsi 2.5 (Hasil *Scan* ke *Database*). Contoh tabel report_detail :

report_id	*Host*	port	type	data	solution	CVE_ID	risk_factor
15	167.205. 49.144	General /tcp	NOTE	Nmap found that this *host* is running Windows NT4/ Win95/ Win98		CAN-1999-0524	Low

6. Tabel report_header

Sedangkan keterangan lain yang menyertai setiap data hasil *scheduled scan* dimasukkan ke tabel ini oleh sistem PANS. *Field* yang menyusun tabel report_header adalah report_id, *client*, *target*, begin_scan, end_scan. Fungsi yang memanfaatkan tabel ini adalah fungsi 2.5. Contoh tabel :

report_id	*client*	*Target*	begin_*scan*	end_scan
15	167.205.49.142	167.205.49.144	300802170000	300802170908

7. Tabel scan_login_detail

Tabel ini berguna untuk menyimpan data detail hasil *scan* yang dilakukan langsung oleh *user login*. *Field* penyusun dan contoh tabel sama seperti terlihat pada tabel 5 (Tabel report_detail). Fungsi yang memanfaatkan adalah fungsi 2.5.

8. Tabel scan_login_header

Keterangan tambahan setiap data hasil *scan* yang dilakukan langsung oleh *user login* disimpan disini. *Field* penyusunnya adalah *field* report_id, idscanner,*target*, scantime, endtime, status, levelscan, *client*. Fungsi yang memanfaatkan adalah fungsi 2.5. Contoh tabel :

report_id	idscanner	*target*	scantime	endtime	status	levelscan	*client*
12	167.205.49.145	167.205.49.144	310802141645	310802142508	1	*Safe*	167.205.49.142

9. Tabel scan_nologin_detail

Sama seperti tabel scan_login_detail hanya saja tabel ini digunakan untuk menyimpan data hasil *scan* yang dilakukan oleh *user nologin*. Contoh tabel seperti pada tabel 5 (Tabel report_detail).

10. Tabel scan_nologin_header

Sama seperti tabel scan_login_header hanya saja tabel ini digunakan untuk menyimpan data hasil *scan* yang dilakukan oleh *user nologin*. Contoh tabel seperti pada tabel 7 (Tabel scan_login_detail).

11. Tabel *schedule*

Semua jadwal *scan* yang telah diatur oleh *user login* disimpan sistem pada tabel *schedule*. Tabel ini terdiri dari *field* menit, jam, tanggal, bulan, hari, cron, mode. Fungsi yang menggunakannya adalah fungsi 3.1. Contoh tabel *schedule* :

menit	jam	tanggal	bulan	hari	cron	mode
0	17	*	*	*	Cron_for_167.205.49.144	perhari

12. Tabel *user*

Sistem PANS menyimpan semua data tentang *user* yang boleh mengakses PANS (*user login*) di dalam tanel ini. Terdiri dari *field* user_id, user_name, real_name, email, *password*. Fungsi yang memanfaatkannya adalah fungsi 1.1-1.5. Contoh tabelnya :

user_id	user_name	real_name	email	*password*
1	admin	Admin PANS	admin@pansserver	7dd66913004434da295aefa937f55c8e

13. Tabel user_login

Pada tabel user_login disimpan data tentang *user* yang sedang *login* pada sistem PANS. *Field* penyusunnya adalah user_name, remote_addr, sesid, logintime, *time*. Fungsi yang menggunakan adalah fungsi 1.1 dan 1.4. Berikut contoh tabel user_login :

user_name	remote_addr	sesid	logintime	*time*
admin	167.205.49.145	9650c1b299abfaa22522c2d839df65fd	1042262627	1042263231

14. Tabel *vulnerabilities*

Data referensi *vulnerability* yang di-*download* dari ICAT *Metabase* (http://icat.nist.gov) disimpan dalam tabel ini. Pada paket *deploy* PANS, sudah disertakan tabel yang berformat sama seperti ICAT *Metabase* yang asli yaitu memiliki total 87 *field*, tetapi yang dimanfaatkan oleh PANS hanya *field-field* berikut : CVE_ID, Publish_Date, CVE_Description, Severity, HL1, HL2, Vuln_Software, EC_Specific_Component. Fungsi yang memanfaatkan tabel ini adalah fungsi 4.2 dan 5.1-5.4. Contoh tabel *vulnerabilities* :

CVE_ID	Publish_Date	CVE_Description	Severity	HL1	HL2	Vuln_Software	EC_Specific_Component
CVE-2002-0153	2002-04-22 00:00:00	Internet Explorer 5.1 for Macintosh allows remote attackers to bypass security checks and invoke local AppleScripts within a specific HTML element…	High	http://www.microsoft.com/technet/security/bulletin/ms02-019.asp	http://online.securityfocus.com/bid/3935	Microsoft Internet Explorer 3.0, 3.1, 4.0, 4.0.1, 4.5, 5.0, 5.1 for MacOS	