IMPLEMENTASI NETWORK ACCESS CONTROL PADA JARINGAN EEPIS

Ali Latiful Aprianto[1], Idris Winarno, SST M.Kom[2]
[1]Mahasiswa Jurusan Teknik Informatika, [2]Dosen Jurusan Teknik Informatika
Jurusan Teknik Informatika Politeknik Elektronika Negeri Surabaya Institut Teknologi Sepuluh Nopember
Kampus ITS Sukolilo, Surabaya 60111, Indonesia
Tel: +62 (31) 594 7280; Fax: +62 (31) 594 6114

ABSTRAK

Belakangan ini, pencurian identitas pribadi melalui media internet semakin marak. Berbagai cara bisa digunakan misalnya melalui phishing, email scan ataupun menggunakan piranti yang sanggup melacak gerak-gerik kebiasaan user ketika mengakses situs-situs web di internet. Kebocoran informasi ini tidak hanya terjadi secara personal tapi juga dapat terjadi secara korporat. Yang mana tidak tertutup kemungkinan kebocoran itu datang dari orang dalam sendiri.

Karena itulah diperlukan adanya pengaman jaringan diantaranya dengan menggunakan metode NAC (Network Access Control). Dengan menggunakan metode NAC, seorang administrator jaringan dapat mengontrol dan mengamankan jaringannya dari aksi para user yang tidak bertanggung jawab dengan cara mengisolasi komputer user tersebut dari koneksi jaringan. Dalam pengerjaan tugas akhir ini, terdapat sebuah komponen tambahan yang digunakan, yaitu SNORT NIDS.

NAC dapat dikembangkan lebih lanjut dengan menambah komponen-komponen pendukung lain. Komponen-komponen tersebut diantaranya adalah hping, nmap, nessus, ethereal dan masih banyak lagi. Dengan NAC ini diharapkan keamanan dalam jaringan akan semakin terjamin.

I. PENDAHULUAN

1.1 LATAR BELAKANG MASALAH

Belakangan ini, pencurian identitas pribadi melalui media internet semakin marak. Utamanya, yang menjadi incaran adalah nomor rekening kartu kredit beserta *password* akun bank dan informasi-informasi penting lainnya. Berbagai cara bisa digunakan misalnya melalui *phishing*, *email scan* ataupun menggunakan piranti yang sanggup melacak gerak-gerik kebiasaan *user* ketika mengakses situs-situs web di internet. Kebocoran informasi ini tidak hanya terjadi secara personal tapi juga dapat terjadi secara korporat. Yang mana tidak tertutup kemungkinan kebocoran itu datang dari orang dalam sendiri.

Karena itulah diperlukan adanya pengaman jaringan diantaranya dengan menggunakan metode NAC (*Network Access Control*). Dimana dengan menggunakan metode NAC, seorang *administrator* jaringan dapat mengontrol dan mengamankan jaringannya dari aksi para *user* yang tidak bertanggung jawab dengan cara mengisolasi komputer *user* tersebut dari sambungan jaringan. Dengan NAC ini diharapkan keamanan dalam jaringan akan semakin terjamin.

1.2 TUJUAN

Pengembangan *Network Access Control* pada dasarnya mempunyai beberapa tujuan, antara lain:

1. Berkurangnya serangan terhadap *server*
2. Pengaturan jaringan secara terpusat
3. Access management

1.3 RUMUSAN MASALAH

Beberapa masalah yang tercakup dalam pembuatan proyek akhir in antara lain adalah:

1. Cara mendapatkan MAC *address* berdasarkan IP secara otomatis sebagai penanda suatu perangkat dalam jaringan.
2. Cara mendeteksi aktivitas abnormal (virus, *spyware*, *worm*) dari suatu perangkat yang terhubung dalam jaringan.
3. Cara mengisolasi suatu perangkat yang terdeteksi melakukan aktivitas abnormal.

1.4 BATASAN MASALAH

Dalam pengerjaan proyek akhir ini terdapat beberapa batasan masalah antara lain:

1. Jaringan yang dimaksudkan adalah suatu jaringan yang terhubung dengan menggunakan suatu switch yang *manageable.*
2. Switch yang digunakan adalah *Cisco Catalys*t yang menggunakan *Cisco* IOS versi12.0 keatas.
3. Kegiatan abnormal jaringan adalah semua pola paket data yang terdefinisi dalam aturan-aturan snort .
4. Perangkat jaringan yang dikelola oleh NAC adalah perangkat yang terdaftar melalui *trap* pada switch saat aplikasi NAC berjalan.

II. TEORI PENUNJANG

Network Access Control (NAC) merupakan sebuah pendekatan dalam keamanan jaringan komputer yang berusaha untuk memadukan beberapa teknologi pengamanan jaringan, seperti antivirus, *host intrusion prevention*, dan otentikasi pada sistem serta keamanan jaringan lainnya.

Network Access Control (NAC) adalah sebuah solusi dalam keamanan jaringan komputer yang menggunakan beberapa protokol untuk mendefinisikan dan mengimplementasikan sebuah aturan yang mendeskripsikan cara untuk mengamankan sebuah akses ke dalam sebuah jaringan ketika sebuah alat mencoba untuk tersambung dalam suatu jaringan.

Sesuai dengan namanya NAC bertujuan untuk mengontrol akses dalam suatu jaringan dengan aturan-aturan tertentu yang telah diatur sebelumnya. Dimana seorang *administrator* dapat menentukan perangkat mana saja yang dapat mengakses suatu jaringan, dan apa yang dapat dilakukan perangkat tersebut dalam suatu jaringan. Sehingga jaringan tersebut terhindar dari serangan virus, *host intrusion*, dan *network worm*.

III. PERANCANGAN SISTEM

3.1 METODE YANG DIGUNAKAN

Beberapa metode atau tahapan yang digunakan pada pengerjaan proyek akhir ini antara lain:

1. *Filtering* Paket Data dengan Snort

Filtering paket data pada proyek akhir ini adalah dengan menggunakan *tool* open source yang bernama snort . Snort merupakan NIDS (*Network Intrusion Detection System*) yang telah digunakan di banyak instansi. Dengan menggunakan snort maka kita dapat mendeteksi kegiatan abnormal pada suatu jaringan. Kegiatan abnormal tersebut dapat disebabkan oleh virus, *spyware*, ataupun *user* itu sendiri. *Output* dari snort dapat dituliskan pada suatu *file* dan atau disimpan pada suatu *database* dalam komputer *server*.

2. Mendapatkan Data Klien dengan SNMP *Trap*

Untuk mendapatkan MAC *address* suatu komputer yang terhubung dengan suatu switch manageable adalah dengan menggunakan SNMP *trap*. SNMP *trap* adalah suatu fasilitas yang tersedia dalam switch yang digunakan untuk mengirimkan sebuah *trap* (pesan) ke komputer *server*. Dalam *trap* tersebut terdapat MAC *address* dari komputer yang tersambung pada switch. *Trap* yang digunakan pada pengerjaan proyek akhir ini adalah:

a. *Link-Up* / *Link-Down Trap*

Link-up adalah sebuah *trap* yang akan aktif secara otomatis saat sebuah komputer tersambung dalam jaringan. Sedangkan *Link-down trap* adalah sebuah *trap* yang aktif ketika komputer terputus dari jaringan.

b. MAC *Notification Trap*

MAC *notification trap* hanya tersedia bagi beberapa switch saja. *Trap* ini adalah sebuah *trap* yang digunakan untuk pendeteksian MAC *address* dari komputer yang terhubung ke switch. Dengan *trap* ini maka komputer *server* tidak memerlukan modul khusus untuk melakukan *query* MAC *address* terhadap switch.

3. Membaca *Output* Snort Secara *Real Time*

Output Snort terbagi menjadi dua yaitu *output* yang terdapat dalam *database*, dan *output* yang terdapat dalam *file*. *Output* snort yang digunakan pada proyek akhir ini adalah output snort dalam bentuk database. Untuk membaca output tersebut dapat dengan menggunakan suatu thread yang membaca *database* tersebut secara terus menerus. Pembacaan *database* dapat dengan melakukan *query* pada *database* tersebut, sehingga data dapat langsung digunakan untuk proses selanjutnya. Data yang didapat dari pembacaan ini nantinya akan digunakan sebagai acuan untuk merubah vlan pada switch.

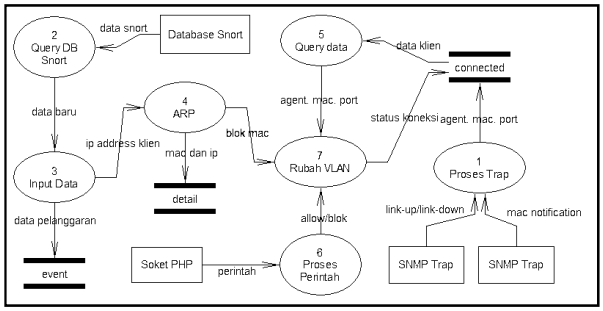
4. Merubah VLAN

Mac address yang terdeteksi melakukan kegiatan abnormal dapat diisolasi dengan cara merubah vlan pada *port* switch yang terhubung pada mac tersebut. Vlan dapat diubah dengan cara menggunakan fasilitas snmp. Untuk melakukan otomasinya maka dapat dengan menuliskan program *socket* yang mengirimkan perintah ke switch melalui *port* 161 untuk mengambil *file* konfigurasi yang ada pada *server* tftp. Suatu *client* yang vlan-nya berubah tidak dapat terkoneksi kembali ke dalam jaringan kecuali *administrator* merubah vlan *port* yang digunakan oleh *client* tersebut pada vlan yang sebelumnya atau vlan normal.

5. Command PHP – Java Melalui *Socket*

Perintah dari bahasa pemrograman PHP dapat dikirimkan ke java dengan menggunakan suatu *socket* yang telah dipersiapkan terlebih dahulu. *Socket server* berada pada java dengan tipe *socket* yang dibuat adalah *socket* TCP. *Socket* ini akan mendengarkan semua perintah yang dikirimkan melalui *socket* dengan *port* tertentu. Apabila perintah yang diterima cocok, maka program java akan melakukan perintah tersebut. *Socket* pada PHP dibentuk dengan menggunakan fungsi fsockopen(). Sedangkan untuk mengirimkan perintah digunakan fungsi fputs().

3.2 PERANCANGAN DATABASE



Gambar 3.12 DFD *Database* TA

Pada gambar 3.12 di atas terlihat adanya tiga buah tabel yaitu tabel event, detail dan connected yang dimodifikasi oleh banyak proses. Proses 1 adalah sebuah proses pengolahan trap yang didapatkan dari switch. Proses 2 adalah suatu proses yang melakukan query pada database snort secara terus menerus. Jika pada proses 2 diketemukan data baru maka data tersebut akan dimasukkan pada tabel event dan tabel detail seperti tampak pada proses 3. Untuk mendapatkan mac address dari ip address yang dimasukkan maka digunakan proses 4 yaitu proses ARP. Mac address yang didapatkan kemudian digunakan untuk merubah vlan pada switch.

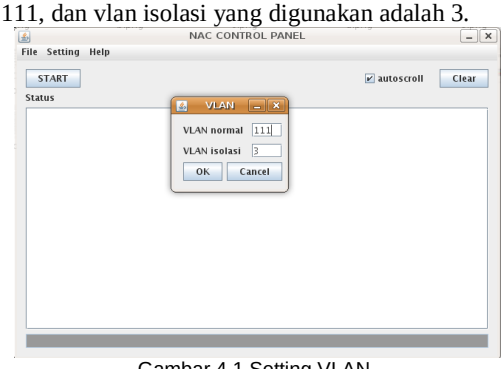
IV. PENGUJIAN DAN ANALISIS

4.1 MENJALANKAN NAC *CONTROL PANEL*

Sebelum menyalakan NAC *control panel*, Snort, *Apache*, MySQL, dan TFTP harus telah menyala dan dapat beroperasi normal. Untuk setting konfigurasi switch yang digunakan pada saat pengujian ini dilakukan dapat dilihat pada lampiran 2. Beberapa hal yang sebaiknya dilakukan sebelum menekan tombol *start* pada NAC control panel adalah:

1. Memasukkan vlan id yang digunakan

Vlan id yang dipakai pada switch dapat dimasukkan melalui sebuah *frame* khusus yang dapat diakses lewat menu *Setting*, VLAN. Seperti tampak pada gambar 4.1, *setting* vlan mempunyai dua masukan. Masukan tersebut adalah vlan normal dan vlan isolasi. Dalam gambar 4.1 diasumsikan bahwa vlan normal yang digunakan adalah vlan 111, dan vlan isolasi yang digunakan adalah 3.

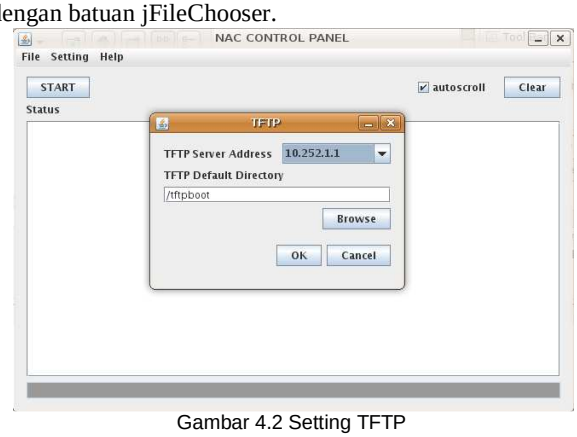


Gambar 4.1 Setting VLAN

Pada *frame* ini *user* harus memasukkan ID vlan yang ingin digunakan. ID vlan yang dimasukkan tergantung pada keinginan *user* namun tetap harus diantara 1 – 1001 (*range* id vlan secara normal). Vlan normal diisi dengan nomor vlan yang digunakan untuk dapat tersambung ke dalam jaringan. Vlan normal yang dimasukkan haruslah sama dengan vlan yang digunakan oleh program NAC. Untuk vlan isolasi, pastikan pada vlan ini merupakan vlan kosong, yaitu vlan yang tidak tersambung ke jaringan.

2. Memasukkan *server* dan direktori TFTP

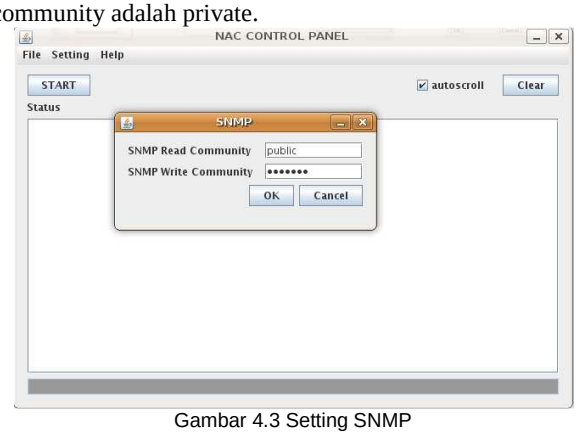
Untuk memasukkan *server address* tftp dan tftp direktori dapat dilakukan dengan memilih pilihan pada menu *Setting*, TFTP. Dalam *frame* tftp seperti tampak pada gambar 4.2 terdapat dua input yang harus dimasukkan. Input yang pertama adalah input *server address* tftp yang dapat dimasukkan dengan memilih ip *address* yang ada pada *combo box*. Pastikan untuk memilih IP yang mempunyai *net-id* yang sama dengan *net-id* pada switch. Dalam pengujian ini ip server NAC yang dipilih adalah 10.252.1.1, karena ip yang digunakan switch adalah 10.252.1.50. Untuk input direktori dapat dilakukan dengan menuliskan nama direktori yang digunakan oleh tftp atau dapat pula dengan memilih *browse*, untuk memilih direktori dengan batuan jFileChooser.



Gambar 4.2 Setting TFTP

3. Memasukkan *read* dan *write community*

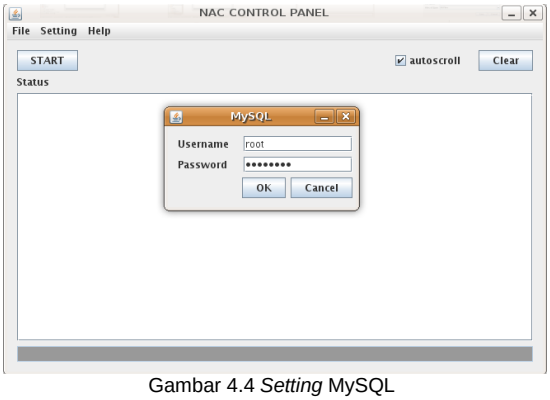
Read dan *write community* merupakan bagian penting dari snmp. *Write community* pada snmp dapat pula dikatakan sebuah *password* yang digunakan snmp untuk dapat merubah konfigurasi yang ada pada switch. *Frame* yang tampak pada gambar 4.3 ini dapat diakses melalui menu *Setting*, SNMP. Pada pengujian sistem ini read community yang digunakan adalah public sedangkan write community adalah private.



Gambar 4.3 Setting SNMP

4. Memasukkan *username* dan *password* MySQL

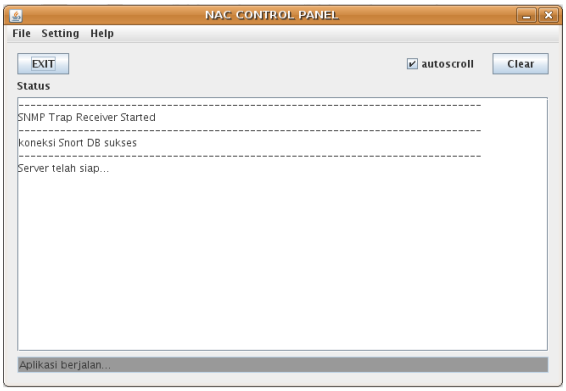
Program NAC ini menggunakan koneksi database MySQL. Karena itu program NAC ini dilengkapi dengan frame input username dan password MySQL. Pada kotak input password setiap karakter yang dimasukkan akan ditutupi dengan karakter asterix seperti tampak pada gambar 4.4.



Gambar 4.4 *Setting* MySQL

5. Start NAC *Control Panel*

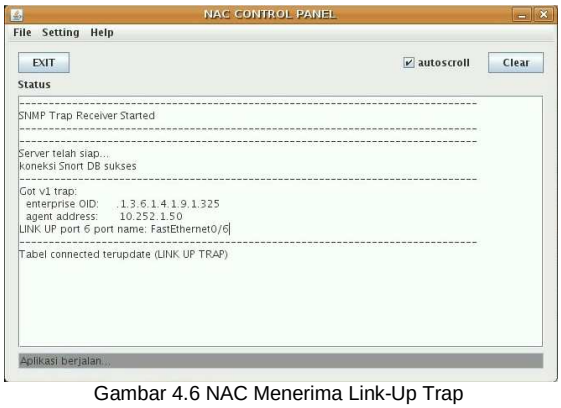
Setelah NAC *Control Panel* terkonfigurasi dengan benar maka aplikasi ini dapat dijalankan dengan menekan tombol *start* yang ada pada *frame* utama. Gambar 4.5 merupakan gambar NAC *Control Panel* yang telah berjalan dengan benar. Pada gambar tersebut terlihat status *trap receiver* siap, koneksi *database* sukses, dan *server* telah siap.



Gambar 4.5 Menjalankan Aplikasi NAC

4.2 MENGKONEKSIKAN KOMPUTER KLIEN

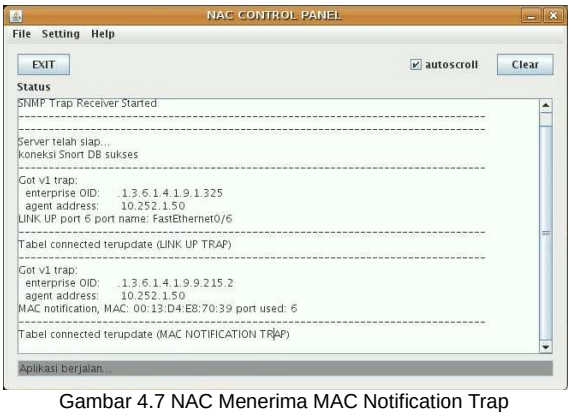
Untuk melakukan koneksi dengan komputer *client* dibutuhkan kabel LAN tipe *straight* (lurus) untuk ditancapkan pada *port* switch dan pada *port* LAN komputer.



Gambar 4.6 NAC Menerima Link-Up Trap

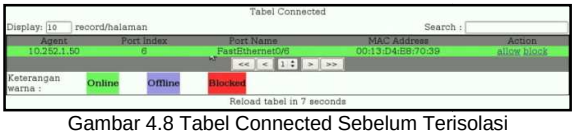
Pada gambar 4.6 terlihat bahwa NAC *server* menerima *trap link-up*. *Trap* ini adalah sebuah *trap* yang dikirimkan oleh switch saat ada suatu alat jaringan yang terkoneksi ke dalam *port* switch. Informasi yang dapat diambil dari tram ini adalah *IP address* switch, indek *port* dan nama *port* yang digunakan. Dapat dikatakan bahwa NAC *Control* Panel telah berhasil mengelola *link-up trap* yang dikirim oleh switch.

Trap lain yang akan dikirim oleh switch saat ada klien terkoneksi adalah *mac-address* notification seperti tampak pada gambar 4.7. Dalam *trap* ini terdapat informasi mengenai mac *address* yang digunakan oleh suatu klien. Dengan menggunakan mac *address* ini NAC *Control* Panel kembali mengirimkan perintah snmp pada switch untuk mengirimkan indeks *port* yang digunakan oleh *mac-address* tersebut.



Gambar 4.7 NAC Menerima MAC Notification Trap

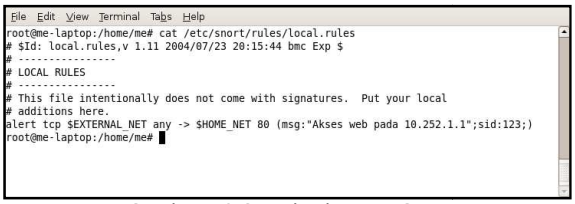
Jika *trap link-up* dan *trap mac-address notification* telah berhasil diproses maka pada GUI NAC yang terdapat pada PHP akan ditambahkan dengan satu baris baru seperti tampak pada gambar 4.8. Dalam baris ini terdapat informasi *trap* yang telah diproses. Informasi tersebut berisi tentang *IP address* switch, *port* indeks, nama *port*, dan *mac-address* klien.



Gambar 4.8 Tabel Connected Sebelum Terisolasi

4.1.2 KLIEN MELAKUKAN KEGIATAN ABNORMAL

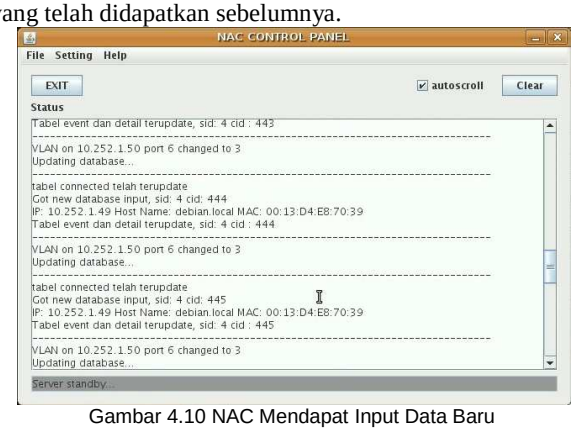
Perlu diingat bahwa yang dimaksud kegiatan abnormal adalah semua paket data yang memiliki pola yang sama dengan pola paket data yang dituliskan dalam aturan snort .



Gambar 4.9 Contoh Isi Aturan Snort

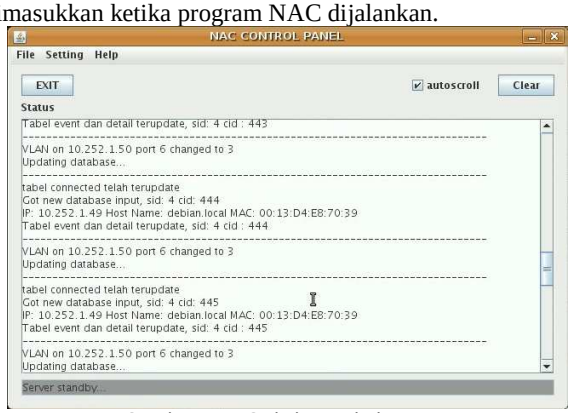
Gambar 4.9 merupakan tampilan dari salah satu aturan snort yang dibuka dengan menggunakan notepad++. Dalam aturan tersebut terlihat ada sebuah aturan yang menyebutkan bahwa setiap klien jaringan yang berusaha mengakses web (*port* 80) pada *ip address* 10.252.1.1 adalah suatu paket data yang perlu di-*filter* oleh snort . Aturan ini sengaja ditulis untuk mencoba apakah NAC *Control Panel* dapat bekerja dengan semestinya yaitu dengan mengisolasi klien yang berusaha mengakses web pada *ip address* 10.252.1.1.

Saat klien melakukan akses web terhadap *IP address* 10.252.1.1 maka secara otomatis oleh snort paket data tersebut akan di-*log* ke *database*. *Log database* snort ini selanjutnya diproses oleh program NAC untuk dicocokkan antara *ip address* yang didapatkan dari *database* snort dengan *mac-address* yang didapatkan dari pemrosesan *trap* yang telah didapatkan sebelumnya.



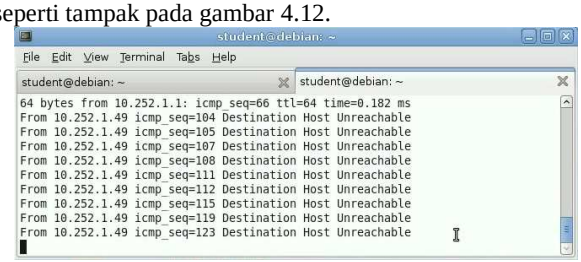
Gambar 4.10 NAC Mendapat Input Data Baru

Dari gambar 4.10 terlihat adanya suatu aktifitas jaringan yang dilakukan oleh klien 10.252.1.7 terjaring oleh snort . Saat suatu aktivitas terjaring oleh snort maka secara otomatis koneksi tersebut akan diputuskan oleh program NAC. Hal ini dapat dilihat pada gambar 4.11, yang menunjukkan bahwa VLAN switch 10.252.1.50 pada *port* 6 diubah menjadi 3, sesuai dengan VLAN isolasi yang dimasukkan ketika program NAC dijalankan.



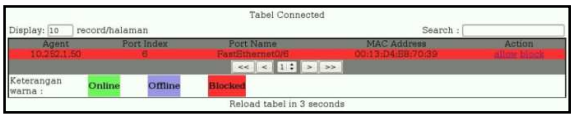
Gambar 4.11 Switch Berubah VLAN

Setelah proses rubah VLAN pada gambar 4.11, klien yang terkoneksi pada *port* 6 sudah terisolasi dari jaringan. Hal ini dapat dibuktikan dengan melakukan perintah *ping* seperti tampak pada gambar 4.12.



Gambar 4.12 Ping Gagal Reply

Klien yang terisolasi oleh NAC, pada php akan diwarnai dengan warna merah seperti tampak pada gambar 4.13.



Gambar 4.13 Tabel Connected Setelah Terisolasi

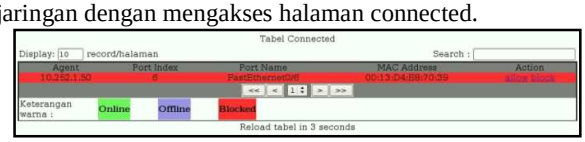
4.1.3 MENGELOLA NAC MELALUI PHP

Aplikasi NAC dalam proyek akhir ini menggunakan PHP sebagai *user interface*-nya yang memungkinkan pengguna dapat mengakses sistem ini dari semua komputer yang terhubung ke jaringan tersebut.

Sebelum dapat melakukan administrasi, seorang *user* akan dihadapkan dengan halaman login seperti pada gambar 4.14. Pada halaman ini *user* diminta untuk memasukkan *user* dan *password*. *Username* yang digunakan saat pengujian program ini adalah root dan *password*-nya adalah root.

Gambar 4.14 Input User dan Password

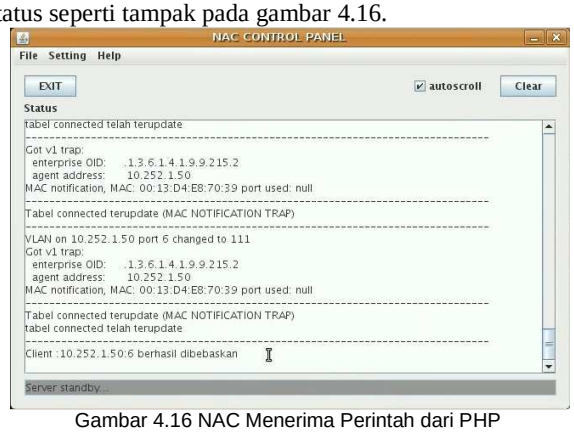
User yang sukses melakukan *login* dapat melakukan control jaringan dengan mengakses halaman connected.



Gambar 4.15 Halaman Connected Sebelum Perintah Allow

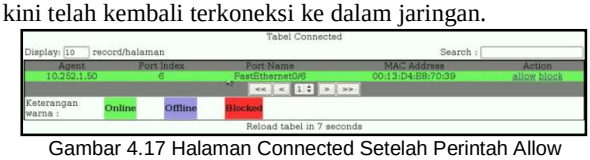
Gambar 4.15 merupakan potongan gambar dari halaman connected. Pada gambar tersebut terlihat adanya suatu klien yang terisolasi oleh NAC. Hal ini dapat diketahui dari warna merah yang mewarnai baris pada tabel tersebut. Untuk merubah VLAN klien yang terisolasi ke VLAN normal dapat dilakukan dengan cara menekan *link allow* yang ada pada kolom *action*. Saat link allow dipilih, PHP akan membentuk sebuah soket yang berhubungan langsung dengan program java yang ada pada *server* NAC, serta mengirimkan *command allow* pada *socket* tersebut.

Program NAC pada java akan menerima perintah tersebut dan memproses perintah itu ke dalam switch. Penggantian VLAN pada switch akan tampak pada kotak status seperti tampak pada gambar 4.16.



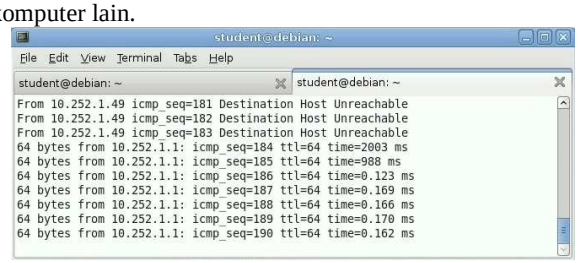
Gambar 4.16 NAC Menerima Perintah dari PHP

Setelah proses pergantian VLAN pada switch sukses dilakukan, warna pada baris tabel halaman connected akan berubah mengikuti status klien tersebut. Seperti tampak pada gambar 4.17, warna berubah dari merah menjadi hijau. Hal ini menunjukkan bahwa klien yang tadinya terisolasi kini telah kembali terkoneksi ke dalam jaringan.



Gambar 4.17 Halaman Connected Setelah Perintah Allow

Untuk memastikan apakah klien telah terkoneksi kembali dapat dilakukan dengan cara menjalankan perintah *ping*. Pada gambar 4.18 terlihat bahwa klien telah tersambung kembali ke dalam jaringan. Hal ini dibuktikan dengan perintah *ping* yang sukses mendapatkan *reply* dari komputer lain.



Gambar 4.18 Ping Berhasil Reply

4.2 KELEBIHAN DAN KELEMAHAN SISTEM

Pada bagian ini akan disebutkan beberapa kelebihan-kelebihan yang ditawarkan program NAC. Pada bagian ini juga akan disebutkan beberapa kelemahan yang masih terdapat dalam proyek akhir ini.

4.2.1 KELEBIHAN SISTEM

Beberapa kelebihan yang ditawarkan oleh sistem NAC ini adalah:

1. Mampu membaca output snort dalam *database* MySQL secara *real-time.*
2. Aplikasi ini dapat menerima *command* dari PHP melalui sebuah *socket.*
3. Mampu menerima dan memproses data *trap* secara *real-time*.
4. Aplikasi ini hanya memerlukan satu NIC *card* untuk mengawasi paket data, menerima snmp *trap*, dan mengirimkan perintah pada switch.
5. Dalam GUI NAC pada PHP dilengkapi dengan sistem ajax, guna mempermudah pemakai dalam hal penggunaan.

4.2.2 KELEMAHAN SISTEM

Beberapa kelemahan yang dijumpai saat percobaan proyek akhir ini adalah:

1. Belum ada modul yang berfungsi untuk mengecek apakah proses snort , *mysql* dan *apache* telah berjalan dengan benar.
2. Tidak adanya enkripsi perintah yang dikirimkan oleh PHP ke Java melalui soket.
3. Aplikasi masih belum memiliki manajemen *error* yang memadai, karena setiap ada *error* yang dijumpai maka program akan dipaksa untuk keluar.
4. Pada aplikasi ini belum ada fitur untuk menyimpan konfigurasi, sehingga setiap program dijalankan ulang maka *setting* vlan, tftp, dan snmp pada program akan kembali ke *setting* awal.

5. Pada interface NAC di PHP terdapat fitur *login*, namun masih belum memiliki manajemen *user* guna menambah atau mengurangi *user* yang dapat melakukan *login*.
6. Tidak memiliki notifikasi klien yang berfungsi untuk memberitahu seorang klien apabila klien tersebut terisolasi dari jaringan.
7. Program NAC ini hanya mendukung satu *read community* dan satu *write community*. Jadi jika dalam jaringan terdapat dua atau lebih switch, maka *read* dan *write community* pada switch tersebut harus sama.

V. PENUTUP

5.1 SIMPULAN

Berdasarkan analisa dari beberapa pengujian yang diterangkan pada bab sebelumnya, kesimpulan yang didapatkan adalah:

1. Aplikasi ini mampu mengubah VLAN secara otomatis maupun secara manual dengan input *user* melalui PHP.
2. Aplikasi ini mampu mempermudah kerja seorang *administrator* jaringan dalam mengawasi serta mengatur hak akses dalam jaringan.
3. Aplikasi ini dapat mengisolasi komputer klien jika *packet* data yang ditransmisikan sesuai dengan *rule* yang telah dikonfigurasikan dalam aturan-aturan snort .
4. Aplikasi ini mampu mengatasi klien yang melakukan koneksi ataupun serangan secara bersamaan.

5.2 SARAN

Hal yang perlu diperhatikan untuk mengembangkan sistem ini lebih lanjut yaitu perlunya mengatasi kelemahan-kelemahan yang telah dituliskan pada bab sebelumnya, serta menambahkan komponen sekuritas yang lain misalnya: hping, nmap, nessus, dan ethereal. Sehingga program NAC ini dapat menjadi program yang simpel dan praktis namun tangguh untuk mengatasi serangan pada jaringan.

DAFTAR PUSTAKA

[1] Fratto Mike, Gaining *Control*, *Network Access Control* Tutorial, web[http://www.networkcomputing.com/channels/security/showArticle.jhtml?articleID=201001835] (28 Desember 2008)

[2] Michael E. Steele, Installing a complete IDS using the *Apache* Web*server*, web[http://www.winsnort .com/index.php?module=Pages&func=display&pageid=11] (2 Maret 2009)

[3] PaketFence–version 1.7.5 Installation and Configuration Guide, web[http://prdownloads.sourceforge.net/packetfence/PacketFence_Installation_Guide-1.7.5.pdf?download] (28 Desember 2008)

[4] icarus (c) Melonfire, *Socket Programming* With PHP, web[http://www.devshed.com/c/a/ PHP/*Socket-Programming*-With-PHP/] (28 Desember 2008)

[5] Cisco.com, Catalyst 2950 and Catalyst 2955 *Switch* Software Configuration Guide, web[http://www.cisco.com/en/US/docs/switches/lan/catalyst2950/software/release/12.1_14_ea1/configuration/guide/scg2950.pdf] (28 Desember 2008)

[6] Cisco.com, How To Copy Configurations To and From Cisco Devices Using SNMP, web[http://www.cisco.com/application/pdf/paws/15217/copy_configs_snmp.pdf] (28 Desember 2008)

[7] Cisco.com, Using SNMP to Find a *Port* Number from a MAC *Address* on a Catalyst *Switch*, web[http://www.cisco.com/warp/public/477/SNMP/mac*to**port*44800.pdf] (28 Desember 2008)