Rancangan dan Implementasi Sistem Monitoring Deteksi Penyusupan Pada Local Area Network (LAN)

J. Priyanto[1], M. Ulfah[1], Ramdani[1] , M. P. Siallagan[1], M. Nasrun[1].
[1] Jurusan Informatika, FT, Jl. Dipati Ukur Bandung

Abstrak

Keamanan jaringan adalah memberikan jalur yang aman antara entitas yang saling bertukar informasi dan untuk menyediakan perlindungan data. Insiden keamanan jaringan komputer adalah suatu aktivitas yang berkaitan dengan jaringan komputer, dimana aktivitas tersebut memberikan implikasi terhadap keamanan. keamanan jaringan sangat diperlukan dalam suatu sistem jaringan komputer, tujuannya adalah agar informasi-informasi yang penting dapat terlindungi dan terjaga keamanannya dari para penyusup dan pengguna yang tidak memiliki hak dan akses terhadap informasi data tersebut.

Salah satu tujuan dari pengelolaan keamanan komputer adalah mempengaruhi perilaku dari user dengan suatu jalan yang akan melindungi sistem informasi dari permasalahan keamanan. Program monitoring membantu organisasi mencapai tujuan ini dengan meningkatkan kemampuan penemuan resiko.

Abstract

Network security is to give peaceful band between entitas which changing over each other information and to provide protection of data. Security computer network incident is an activity related to computer network, where the activity give implication to security. network security very needed in an computer network system, its target is so to be important informations can protect and awake its security from all consumer and infiltrator which do not have rights and access to data information.

One of the intention of management of computer security is to influence behavior of user with to protect information system of problems of security. Monitoring program assist tired organization of this target by improving ability of invention of risk.

Kata kunci : Keamanan Jaringan, aglets, monitoring

1. Pendahuluan

Sejak Jaringan komputer terus mengalami perkembangan, baik dari skalabilitas, jumlah node dan teknologi yang digunakan. Hal ini memerlukan pengelolaan jaringan yang baik agar ketersediaan jaringan selalu tinggi. Tugas pengelolaan jaringan yang dilakukan administrator jaringan memiliki banyak permasalahan, diantaranya yang berkaitan dengan keamanan jaringan komputer.

Penyusupan adalah seseorang yang berusaha merusak atau menyalahgunakan sistem, atau setiap usaha yang melakukan *compromise* integritas, kepercayaan atau ketersediaan suatu sumberdaya komputer. Definisi ini tidak bergantung pada sukses atau gagalnya aksi tersebut, sehingga berkaitan dengan suatu serangan pada sistem komputer.

2. Pembahasan

Keamanan jaringan adalah memberikan jalur yang aman antara entitas yang saling bertukar informasi dan untuk menyediakan perlindungan data. Insiden keamanan jaringan komputer adalah suatu aktivitas yang berkaitan dengan jaringan komputer, dimana aktivitas tersebut memberikan implikasi terhadap keamanan. keamanan jaringan sangat diperlukan dalam suatu sistem jaringan komputer, tujuannya adalah agar informasi-informasi yang penting dapat terlindungi dan terjaga keamanannya dari para penyusup dan pengguna yang tidak memiliki hak dan akses terhadap informasi data tersebut.

Program monitoring merupakan software yang melakukan otomatisasi proses monitoring kejadian yang muncul di sistem komputer atau jaringan, menganalisanya untuk menemukan permasalahan keamanan.

Local Area Network (LAN) adalah sejumlah komputer yang saling dihubungkan bersama di dalam satu areal tertentu yang tidak begitu luas, seperti di dalam satu kantor atau gedung. Secara garis besar terdapat dua tipe jaringan atau LAN, yaitu jaringan *Peer to Peer* dan jaringan *Client-Serve*r. Pada jaringan peer to peer, setiap komputer yang terhubung ke jaringan dapat bertindak baik sebagai *workstation* maupun *serve*r. Sedangkan pada jaringan *Client-Serve*r, hanya satu komputer yang bertugas sebagai *server* dan komputer lain berperan sebagai *workstatio*n.

Topologi Jaringan adalah gambaran secara fisik dari pola hubungan antara komponen-komponen jaringan, yang meliputi server, workstation, hub dan pengkabelannnya. Terdapat tiga macam topologi jaringan umum digunakan, yaitu Bus, Star dan Ring.

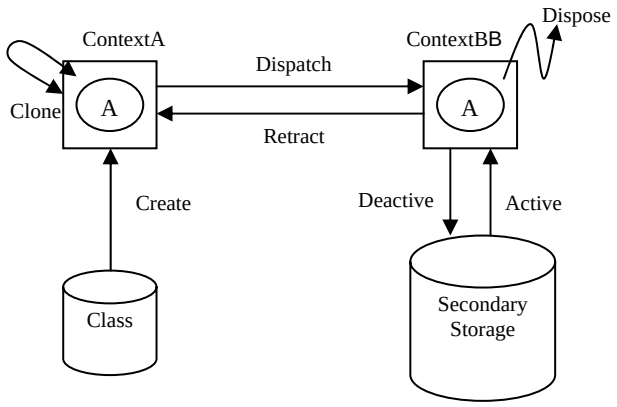
Software Agent (selanjutnya di sebut Agen saja) adalah entitas perangkat lunak yang didedikasikan untuk tujuan tertentu. Agen bisa memiliki ide sendiri mengenai bagaimana menyelesaikan suatu pekerjaan tertentu atau agenda tersendiri.

Mobile agent adalah agen yang aktif dan dapat bergerak menuju komputer lain, atau mejelajahi jaringan untuk menjalankan tugasnya. *Mobile agent* sering digunakan untuk mengumpulkan data, informasi atau suatu perubahan.

Aglets adalah obyek *java* yang dapat bergerak dari satu *host* ke *host* lain dalam suatu jaringan. *Aglet* yang sedang bekerja disuatu *host* dapat menghentikan eksekusinya, pergi ke *host* yang lain dan kemudian memulai eksekusinya kembali. Ketika aglet bergerak, aglet membawa kode program dan juga *state* dari semua obyek yang membawanya. Sebuah mekanisme keamanan yang *built-in* akan mengamankan *host* dari *untrusted aglet*. Istilah *aglet* sesungguhnya adalah kombinasi dari kata *agen* dan *applet*. Perbedaan dengan *applet* adalah *aglet* juga membawa serta *state*, dan memiliki *itinerary* (rencana perjalanan) ASDK (*Aglets Software Development Kit*) adalah paket perangkat lunak yang digunakan untuk menulis aplikasi *mobile agent*. ASDK menggunakan bahasa *java* dan bisa didapatkan secara gratis dari internet. Karena kemampuannya membuat *lightweight agent* , maka *aglet* API ini sering disebut sebagai RISC *mobile agent*.

Tahiti adalah suatu program aplikasi yang bekerja sebagai *server*

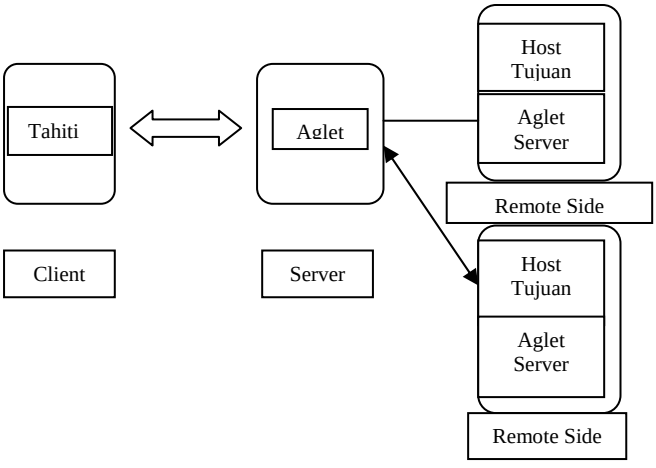
aglet. Aplikasi tahiti merupakan suatu paket dengan ASDK. Beberapa *server* (*tahiti*) dapat dijalankan dalam satu komputer dengan cara memberikan nomor *port* yang berbeda.



Gambar 2.4 Model Siklus Hidup *Aglet*

Proses Perancangan Sistem

Secara blok diagram, perancangan terdiri dari 3 blok utama, yaitu *client side*, *server side*, dan *remote site*. *Client site* berupa *Tahiti* sebagai antarmuka dengan *user*. *Server side* terdiri dari *aglet server* di *host* asal. Sedang *remote site* berupa *aglet server* pada *host* tujuan yang menjadi tujuan deteksi penyusupan serta informasi kondisi *host* tersebut yang akan diambil.



Gambar 3.1 Diagram Blok Arsitektur Sistem

Informasi Deteksi Penyusupan

Sejumlah kegiatan yang akan berusaha dideteksi dan dicurigai sebagai penyusupan bergantung pada kebijakan jaringan yang ditentukan *administrator* :

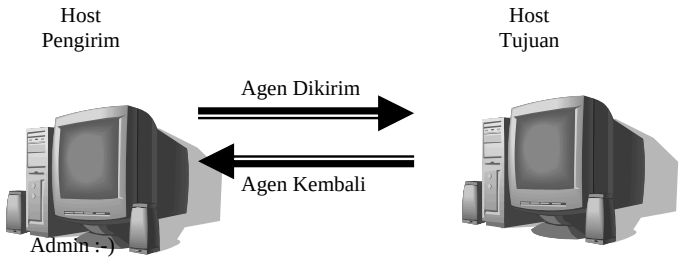
Sejumlah informasi yang diambil:
1. Nama User name *host*
2. Nama Organization
3. Alamat Atp (Agen Transfer Protocol) yang aktif
4. File path yang aktif
5. *Port* tertentu yang aktif
6. *Class* name yang aktif

Penentuan Batasan Sistem

Deteksi penyusupan dilakukan dengan mengirimkan agen melalui jaringan dari satu *host* ke *host* tujuan yang berada dalam jaringan komputer. Agar proses pengiriman agen tidak menghabiskan banyak waktu dan *bandwith* jaringan, agen tersebut harus berukuran cukup kecil, hanya beberapa puluh kilobyte saja. Bahasa pemrograman yang cocok untuk keperluan tersebut adalah *Java* karena hasil kompilasi *Java* cukup kecil dan *Java* sangat handal untuk aplikasi jaringan.

Syarat *host* yang akan menjadi tujuan pengiriman *mobile agent* adalah :
1. Telah di kenal alamat IP-nya dan alamat IP tersebut dapat diakses melalui jaringan.
2. Telah dipasang JDK (*Java Development Kit*) versi 1.1.8 serta *aglet viewer Tahiti* berjalan dengan nomor *port* yang telah ditentukan.
3. Sistem operasi menggunakan NT/2000/XP atau Linux untuk syarat terakhir, yaitu mengenai sistem operasi yang digunakan oleh *host* yang akan menjadi tujuan pengiriman *mobile agent*, perlu sedikit modifikasi pada salah satu *class* dari *mobile agent* bila menggunakan sistem operasi selain tersebut di atas.

Skema Dasar Proses Deteksi Penyusupan



Gambar 3.2 Skema Dasar Deteksi

Langkahnya sebagai berikut :
1. Langkah pertama adalah *Admin* menentukan *host-host* mana yang akan dideteksi.
2. Melakukan *scanning* nama *host* yang aktif. Untuk mempercepat proses, terlebih dahulu dilakukan pengiriman paket *ping* ke masing-masing *host* untuk memeriksa apakah *host* yang bersangkutan sedang aktif atau tidak. Bila sampai batas waktu tertentu tidak diperoleh balasan (*reply*) dari *host* yang bersangkutan. Maka dapat disimpulkan bahwa *host* tersebut sedang tidak aktif atau sedang tidak terhubung ke jaringan sehingga tidak dilakukan proses pengiriman *mobile agent*. Hanya terhadap *host-host* yang sedang aktif saja proses tersebut dilakukan.
3. Mengirimkan agen ke *host* tersebut. Agar agen dapat sampai ketujuan, sebelumnya harus diketahui terlebih dahulu alamat IP atau nama *host* dari *host* tujuan.
4. Di tiap *host* yang disinggahi, agen akan melakukan pengambilan

informasi yang diperlukan untuk

5. *Mobile agent* membawa rencana perjalanannya (*itinerary*) dan akan berpindah ke *host* berikutnya dalam rencana perjalanan tersebut.
6. Setelah semua *host* dalam rencana perjalanan dikunjungi, agen kembali ke *host* pengirim dengan membawa serta informasi yang telah dikumpulkan tersebut.
7. Untuk kemudian dilakukan analisa dan ditampilkan hasilnya di komputer pengirim agen. Informasi yang ditampilkan dalam bentuk informasi setiap *host*, maupun informasi penting seluruh *host* pada jaringan.

Perancangan Agen Program Aplikasi Monitoring

1. Agen *Master*

   Agen Master merupakan agen utama yang akan dijalankan pertama kali oleh user dari antarmuka *Tahiti*. *Agen Master* merupakan *stationary agent* yang mengontrol masukan dari *user*, serta bertugas membentuk, mengirimkan atau menerima informasi dari *agen slave*. Karena ada kemungkinan tidak semua *host* tujuan sedang aktif, maka *user* harus memeriksa *host* yang aktif. Selanjutnya pengiriman *mobile agent* hanya akan ditujukan pada *host* yang aktif saja. Hal ini bertujuan untuk mempercepat proses deteksi. Penentuan tujuan host yang akan di monitoring ditentukan di agen master.
2. Agen *Slave*

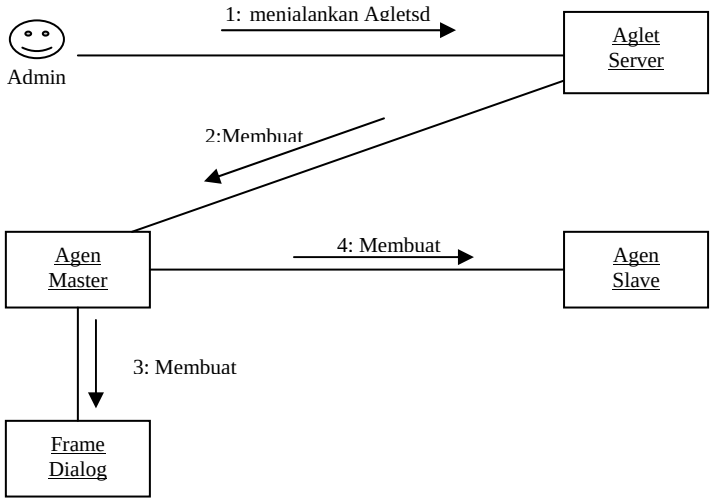
   Ini adalah *agen* yang akan berkeliling jaringan menuju host tujuan. agen ini dibuat oleh agen *Master*, dikirimkan ke *host* tujuan, mengambil informasi dari *host*, dan melanjutkan perjalanan sesuai rencana perjalanan (*itinerary*).

Sejumlah pekerjaan yang dilakukan oleh *agen Slave*:

deteksi penyusupan.

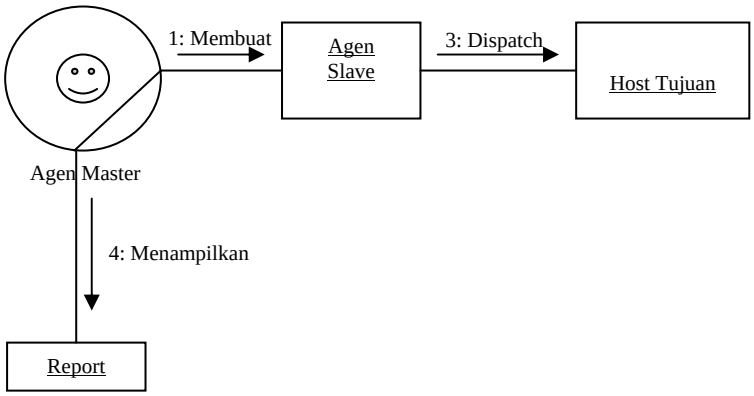
1. Melakukan inisialisasi yang diperlukan untuk memulai perjalanan, seperti rencana perjalanan.
2. Melakukan perjalanan ke *host-host* tujuan.
3. Mengambil informasi pada *host* tersebut.
4. Kembali ke *host* asal pengiriman, memberikan pesan kepulangan ke agen *Master*.

Cara Kerja Agen Dalam Jaringan Komputer



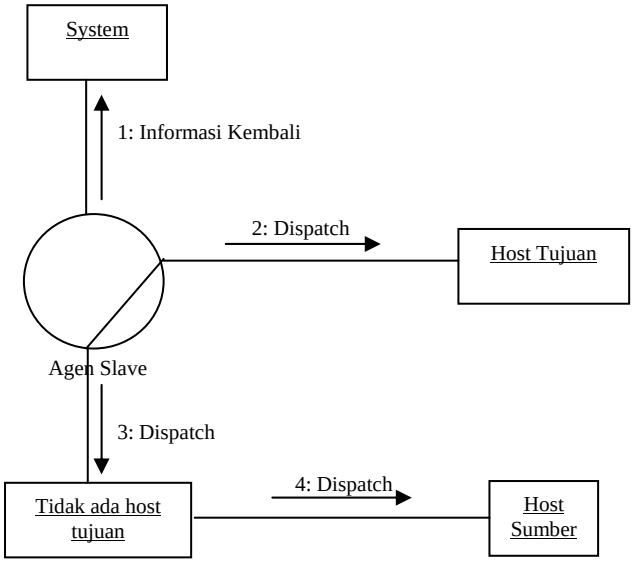
Gambar 3.3 Diagram Pembuatan Gabungan Agen

Selanjutnya agen *Slave* konfigurasi akan diperlukan dalam menentukan pengambilan informasi dan kebijakan jaringan. Lalu *dispatch* agen *Slave* lewat jaringan ke *host* tujuan. Langkahnya dapat dilihat pada diagram kolaborasi di Gambar 4.3.



Gambar 3.4 Diagram Gabungan Dispatch Agen

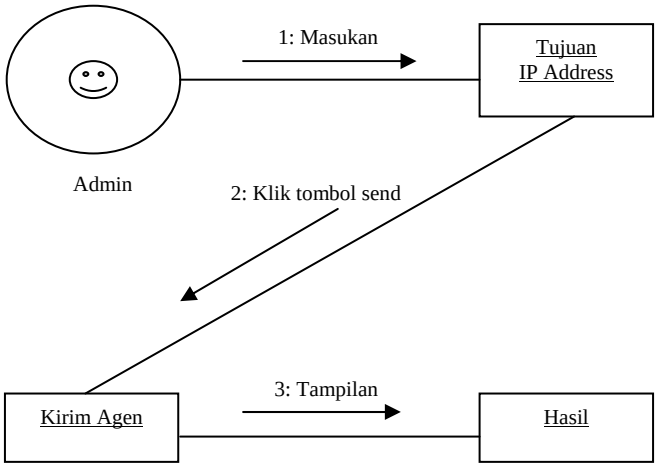
Sesampai di *host* tujuan, agen *Slave* akan mengumpulkan informasi deteksi penyusupan yang diperlukan. Bila tidak ada lagi *host* tujuan maka agen *Slave* akan kembali ke *host* asal pengiriman.



Gambar 3.5 Diagram Gabungan Deteksi Agen

Antarmuka User

Antarmuka *user* disini selain sudah disediakan oleh *Tahiti*, juga dilakukan oleh agen *Master* dengan membuat *frame* dialog. Ini memungkinkan *user* untuk memasukan alamat dari *host-host* yang menjadi tujuan deteksi penyusupan. Kemudian di akhir perjalanan, agen *master* akan menampilkan hasil dari pengumpulan dan analisa informasi deteksi penyusupan yang diperoleh



Gambar 3.6 Diagram kolaborasi User

Analisa Monitoring Menggunakan Agen

Meskipun kenyataannya hampir semua masalah dalam komputasi monitoring dapat diselesaikan tanpa menggunakan *Mobile agent*, namun dengan menerapkan *Mobile agent* akan dapat memudahkan pengembangan aplikasi dan dapat meningkatkan keandalan dan efisiensi. Sedikitnya ada beberapa keuntungan yang didapatkan dari *Mobile agent* :

1. Mengurangi beban jaringan: Sistem terdistribusi sangat tergantung pada protokol komunikasi yang melibatkan banyak interaksi untuk menyelesaikan tugas yang diberikan. Akibatnya *traffic* jaringan tinggi. Sebaliknya, *Mobile agent* memungkinkan untuk mengemas suatu aplikasi, mengirimkannya ke *host* tujuan dan kemudian interaksi dapat terjadi secara lokal. *Mobile agent* juga bermanfaat untuk mengurangi aliran data pada jaringan. Jika terdapat data dengan jumlah yang sangat besar tersimpan di *remote*

host, maka lebih baik memproses data tersebut secara lokal ditempatnya, dari pada mengirimkannya melalui jaringan. *moto mobile agent* adalah: lebih baik memindahkan komputasi ke temapat data, dari pada memindahkan data ke tempat komputasi.

2. Efisiensi sumber daya : Konsumsi sumber daya (CPU dan memori) dapat dihemat, sebab *Mobile agent* menetap dan bekerja hanya pada satu *node* pada satu waktu. *Node* yang lain tidak menjalankan agen sampai *node* tersebut memerlukannya.
3. Menanggulangi *latency* jaringan: Sistem *real-time* yang kritis perlu tanggap terhadap perubahan lingkungannya secara *real-time*. Keterlambatan tanggapan yang diakibatkan oleh masalah jaringan harus dihindari *mobile agent* menawarkan suatu pemecahan dengan mengirimkan agen ke tujuan dan dieksekusi secara lokal.
4. *Encapsulate protocol* : Ketika terjadi pertukaran data pada sistem terdistribusi, setiap *host* memiliki kode sebagai implementasi dari *protocol* yang diperlukan untuk mengkode data yang keluar dan menerjemahkan data yang masuk secara tepat. Untuk menjamin kompatibilitas, kode-kode protokol tersebut sudah dibakukan. Tetapi untuk mengakomodasi keperluan efisiensi dan keamanan yang lebih baru, tidak praktis bila harus meng-*upgrade* kode protokol tersebut, meskipun hal tersebut meungkinkan. *mobile agent* dapat memecahkan masalah tersebut, karena *mobile agent* dapat dikirim ke *remote host* dengan tujuan untuk membentuk *iechannelln* yang berdasarkan protokol yang sudah baku tersebut.
5. Eksekusi secara *asynchronous* dan *autonomou*s : Perangkat *mobile* sering kali harus menggunakan koneksi jaringan yang mahal dan mudah putus. Pekerjaan dapat digabungkan ke dalam *mobile agent* yang kemudian dapat dikirimkan ke tujuan. Setelah dikirimkan dan sampai ke tujuan, koneksi antara perangkat *mobile* dan jaringan dapat diputuskan. *mobile agent* menjadi berdiri sendiri dan dapat beroperasi secara *asynchronous* dan *autonomou*s. Kemudian, perangkat *mobile* dapat dikoneksikan ke jaringan untuk mengambil dan mengumpulkan kembali agen yang telah dikirimkan. Dengan cara *disconected operation* ini *agent dispatched* tidak memerlukan sistem asal untuk tetap aktif sementara mereka dieksekusi. Mereka juga bisa menunggu *(work while you slee*p).
6. Beradaptasi secara dinamis : *mobile agent* mempunyai kemampuan untuk mengetahui perubahan di lingkungan eksekusinya dan dapat bereaksi secara *autonomous* melakukan perubahan. *Multiple mobile agent* mempunyai kemampuan yang khas untuk berdistribusi sendiri di antara *host-host* dalam jaringan sedemikian rupa untuk memelihara konfigurasi yang optimal untuk penyelesaian masalah khusus.
7. Andal dan toleran terhadap kesalahan : Kemampuan *mobile agent* untuk berinteraksi secara dinamis pada situasi dan keadaan yang tak menguntungkan menjadikan *mobile agent* mudah untuk membuat sistem terdistribusi yang andal dan toleran terhadap kesalahan. Jika sebuah *host* akan di-*shutdown* , semua agen yang sedang eksekusi di mesin tersebut akan diberi peringatan dan

diberikan waktu untuk berpindah dan melanjutkan operasinya di *host* yang lain dalam jaringan tersebut.

8. Mendukung lingkungan yang heterogen : Komputasi jaringan pada dasarnya sangat heterogen. Baik pada sisi perangkat keras maupun perangkat lunak. *mobile agent* tidak tergantung pada komputer dan jaringan, tetapi hanya bergantung pada lingkungan eksekusinya. Sebagai contoh, *mobile agent java* dapat ditujukan ke segala sistem yang mempunyai JVM (*Java Virtual Machine*).
9. *Real Time Notification* : Agen yang ditempatkan pada *remote site* bisa memberitahukan perubahan isi secara cepat. Tanpa harus diminta oleh *user*. Ini juga terkait dengan hasil pencarian yang *up to date*, meningkatkan kemampuan penyimpanan data terpusat yang telah ada.
10. Eksekusi parallel : Komputasi masih bisa dibagi ke dalam sejumlah agen, dilakukan *dispatch* ke *node* yang paling sesuai untuk eksekusi dari tiap komponen, dan merakitnya di *home*. Sumber daya *hardware* bukan lagi batasan.
11. Paradigma Komputasi yang adaptif : *mobile agent* bisa di-*retract*, *dispatch*, *clone* atau *deactivate* sesuai dengan perubahan kondisi.
12. Skalabilitas : Bila jumlah elemen komputasi di jaringan meningkat, agen baru bisa dikirim atau *clone* dan *dispatch* ke mesin baru di jaringan.

Kesimpulan

Kesimpulan yang didapatkan dari program aplikasi monitoring menggunakan agen yaitu : Program aplikasi monitoring dapat dipakai sebagai alat mendeteksi suatu host dalam jaringan komputer. Kemampuan untuk memonitoring memungkinkan keamanan data pada host dapat terjaga keamanannya. Apabila terjadi penyalahgunaan data maka program aplikasi monitoring menggunakan agen akan memberitahukannya kepada administrator. Data informasi yang diperoleh dari proses monitoring dapat di jadikan sebagai acuan untuk menetapkan kebijakan keamanan jaringan.Program aplikasi monitoring menggunakan agen mempunyai kemampuan unik untuk memindahkan dirinya sendiri dari satu sistem ke sistem yang lain dalam suatu jaringan.Kemampuan untuk berjalan dijaringan memungkinkan program aplikasi monitoring menggunakan agen untuk berpindah ke sistem yang mengandung obyek yang akan berinteraksi dengan agen di host lainnya dan kemudian mengambil manfaat selama berada dalam *host* dan jaringan yang sama dengan obyek. Program aplikasi monitoring menggunakan agen adalah sebuah program yang dapat menghentikan eksekusi, berjalan melalui jaringan dengan membawa kode dan *stat*e-nya, dan kemudian melanjutkan eksekusinya pada *host* selanjutnya. Program aplikasi monitoring menggunakan agen merupakan teknologi yang menjanjikan untuk implementasi *tool* yang beroperasi pada sumber data pada jaringan komputer.

Daftar Pustaka

1. R.Kresno Aji, Kejahatan Internet, *Trik Aplikasi & tip Penanggulangannya*., Elexmedia Komputindo,2002
2. Rebecca Bace and Petter Mell, Intrusion Detection System, NIST Special Publication on IDS,2002.
3. InfoLinux, *Sistem Pendeteksian Intrusi*,www.infolinux.web.id, juni 2002
4. Sidiq , *Notifikasi dan Akses Database Terdistribusi menggunakan Agen*:, Tesis,2002
5. Internet Security Systems, *Strategy For A Succesfull IDS Evaluation*,www.iss.net,2002
6. Internet Security Systems, *Network VS Host-based Intrusion Detection: A Guide to Intrusion Detection Technology*, www.iss.net.net,2002
7. Jai Sundar Balasubramaniyan, Jose Omar, David Isacoff, dan Diedo Samboni, *An Architecture for Intrusion Detection Using Autonomous Agents*, Center for Education and Research in Information Assurance and Security, Departemen of Computer Sciences Purdue University, 11 juni 1998
8. Mike Bursell, *Aglets Puppies Workshoply*, December 1997
9. Dan Farmer and Wietse Venema,*Improving The Security of Your Site by Breaking Into It,* Sun Microsystem, 1994
10. Craig Hunt, *TCP/IP Network Adminitrationlv*, O™Reilly & Associates, Inc,1992