MEMBANGUN KEAMANAN JARINGAN KOMPUTER DENGAN SISTEM DE-MILITARISED ZONE (DMZ)

Oleh : Kundang K Juman
kundang_kj@yahoo.com

ABSTRAK

DMZ adalah suatu area bagi *hackers* yang digunakan untuk melindungi system internal yang berhubungan dengan serangan *hacker* (*hack attack*). DMZ bekerja pada seluruh dasar pelayanan jaringan yang membutuhkan akses terhadap jaringan " Internet atau dunia luar" ke bagian jaringan yang lainnya. Dengan begitu, seluruh "*open port*" yang berhubungan dengan dunia luar akan berada pada jaringan, sehingga jika seorang *hacker* melakukan serangan dan melakukan *crack* pada *server* yang menggunakan sistem DMZ, *hacker* tersebut hanya akan dapat mengakses *host*nya saja, tidak pada jaringan internal. Secara umum DMZ dibangun berdasarkan tiga buah konsep, yaitu : NAT (*Network Address Translation*), PAT (*Port Addressable Translation*), dan *Access List.* NAT berfungsi untuk menunjukkan kembali paket-paket yang datang dari "*real address*" ke alamat internal. Misal : jika kita memiliki *"real address*" 202.8.90.100, kita dapat membentuk suatu NAT langsung secara otomatis pada data-data yang datang ke 192.168.100.4 (sebuah alamat jaringan internal). Kemudian PAT berfungsi untuk menunjukan data yang datang pada *particular port*, atau *range* sebuah *port* dan *protocol* (TCP/UDP atau lainnya) dan alamat IP ke sebuah *particular port* atau *range* sebuah *port* ke sebuah alamat internal IP. Sedangkan *access list* berfungsi untuk mengontrol secara tepat apa yang datang dan keluar dari jaringan dalam suatu pertanyaan. Misal : kita dapat menolak atau memperbolehkan semua ICMP yang datang ke seluruh alamat IP kecuali untuk sebuah ICMP yang tidak diinginkan.

Kata Kunci : NAT, *real address*, PAT, *Access List*, *Port*, *Protokol*, DMZ, ICMP

PENDAHULUAN

De-Militarised Zone(DMZ) merupakan mekanisme untuk melindungi sistem internal dari serangan hacker atau pihak-pihak lain yang ingin memasuki sistem tanpa mempunyai hak akses. Sehingga karena DMZ dapat diakses oleh pengguna yang tidak mempunyai hak, maka DMZ tidak mengandung rule. .Secara esensial, DMZ melakukan perpindahan semua layanan suatu jaringan ke jaringan lain yang berbeda. DMZ terdiri dari semua port terbuka, yang dapat dilihat oleh pihak luar. Sehingga jika hacker menyerang dan melakukan *cracking* pada *server* yang mempunyai DMZ, maka *hacker* tersebut hanya dapat mengakses host yang berada pada DMZ, tidak pada jaringan internal. Misalnya jika seorang pengguna bekerja di atas server FTP pada jaringan terbuka untuk melakukan akses publik seperti akses internet, maka hacker dapat melakukan cracking pada server FTP dengan memanfaatkan layanan *Network Interconnection System* (NIS), dan *Network File System* (NFS). Sehingga hacker tersebut dapat mengakses seluruh sumber daya jaringan, atau jika tidak, akses jaringan dapat dilakukan dengan sedikit upaya, yaitu dengan menangkap paket yang beredar di jaringan, atau dengan

metoda yang lain. Namun dengan menggunakan lokasi server FTP yang berbeda, maka hacker hanya dapat mengakses DMZ tanpa mempengaruhi sumber daya jaringan yang lain. Selain itu dengan melakukan pemotongan jalur komunikasi pada jaringan internal, trojan dan sejenisnya tidak dapat lagi memasuki jaringan.Makalah ini akan membahas bagaimana memberi hak pada pengguna baik internal maupun eksternal, pada semua layanan jaringan yang diperlukan.

Konsep NAT, PAT, dan Daftar Akses

Network Address Translation(NAT) berfungsi untuk mengarahkan alamat riil, seperti alamat internet, ke bentuk alamat internal. Misalnya alamat riil 202.8.90.100 dapat diarahkan ke bentuk alamat jaringan internal 192.168.0.1 secara otomatis dengan menggunakan NAT. Namun jika semua informasi secara otomatis ditranslasi ke bentuk alamat internal, maka tidak ada lagi kendali terhadap informasi yang masuk. Oleh karena itu maka muncullah PAT.

Port Address Translation(PAT) berfungsi untuk mengarahkan data yang masuk melalui port, sekumpulan port dan protokol, serta alamat IP pada port atau sekumpulan post. Sehingga dapat dilakukan kendali ketat pada setiap data yang mengalir dari dan ke jaringan.

Daftar Akses melakukan layanan pada pengguna agar dapat mengendalikan data jaringan. Daftar Akses dapat menolak atau menerima akses dengan berdasar pada alamat IP, alamat IP tujuan, dan tipe protokol.

Contoh Studi Kasus

Pada sebuah PT.DZX terdapat jaringan komputer berbasis Microsoft Windows Windows NT4.0 untuk mengakses internet, dan Microsoft Exchange 5.5 untuk mail lokal maupun global. Masalah-masalah yang dapat diamankan dengan menggunakan DMZ adalah sebagai berikut:

Semua alamat *Internet Protocol*(IP) merupakan alamat komputer sesungguhnya, sehingga dapat diakses secara langsung dari internet,

Server *Domain Name Server*(DNS) eksternal dapat digunakan pada jaringan internal,

Server Web bekerja di lingkungan internal,Terdapat server *File Transfer Protocol*(FTP) yang bekerja di lingkungan internal,Server Exchange dapat diakses secara langsung dari internet,

Tidak terdapat batasan pada permintaan yang masuk dan keluar jaringan.

Sebaran IP Baru dan memindahkan Layanan Web

PT.DZX juga didukung dengan server RedHat Linux dan dilengkapi dengan kartu ISDN. Semua routing pada server ini di non-aktifkan dan hanya berfungsi sebagai gateway aplikasi yang bekerja dengan melakukan monitoring pada port-port tertentu, dan mengaktifkan program lain yang dapat melayani arus informasi pada jaringan internal. Langkah pengamanan pertama yang dilakukan adalah dengan membenahi alamat IP sehingga dapat digunakan sebagai alamat global. Jika terdapat serangan hacker, maka jaringan internal tidak akan terganggu.Lakukan setup DNS pada Windows NT4.0, karena layanan DNS pada NT relatif mudah dikonfigurasi, cukup aman untuk DNS internal, dan mendukung registrasi dinamis. Versi terbaru dari BIND mendukung registrasi dinamis untuk upgrade ke Windows 2000, sehingga sistem membutuhkan layanan DNS Windows 2000 untuk ekstensi direktori aktif.Kemudian dilakukan modifikasi

pada semua alamat IP pada Server dan Print Server, mengubah konfigurasi aplikasi gateway pada Linux, dan membentuk sebaran DHCP baru. Langkah berikutnya adalah emindahkan halaman web dari jaringan lokal ke ISP karena halaman page tidak harus diubah setiap saat. Dengan memindahkan halaman web ke ISP, maka aspek keamanan bukan menjadi kompleksitas programmer namun menjadi kompleksitas ISP.

Menentukan Perangkat Keras Pendukung

Perangkat keras yang digunakan meliputi koneksi ADSL, implementasi Firewall, dan implementasi DMZ. Pada perangkat keras yang digunakan menggunakan sistem operasi Windows atau Linux. Windows mempunyai kelemahan :
Meskipun Windows NT/2000 cukup sulit di hack, namun mudah diserang Denial Of Service (DOS) atau service yang crash. Banyak sekali pihak yang melakukan hack pada lingkungan Windows. Sedangkan kelemahan Linux adalah karena Linux merupakan sistem operasi yang dibangun oleh hacker sehingga source code Linux mudah didapat. Oleh karena itu dengan menggunakan Linux, maka tingkat keamanan semakin rendah.Perangkat keras yang dibutuhkan terdiri dari perangkat komputer beserta paket sekuritinya, koneksi ADSL dan firewall, serta switch layer Data Link.

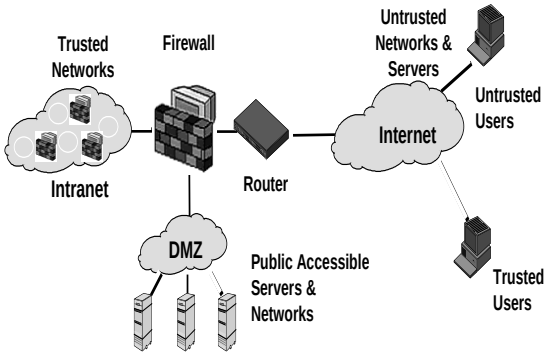
Implementasi Jalur ADSL dan Firewall PIX

Setelah perangkat keras tersedia, maka berikutnya adalah melakukan pemetaan alamat perangkat keras, misalnya: ADSL – 209.15.20.34 Ethernet0 pada ADSL – 192.1.10.5/30 (255.255.255.252)

Ethernet0 pada firewall PIX.Berikutnya dibangun translasi NAT untuk melakukan panggilan forward ke 192.168.10.6. Biarkan router menjadi data route, dan biarkan Firewall menentukan konfigurasi yang diperlukan untuk pengelolaan resiko.
Firewall merupakan sistem yang menyediakan konektifitas yang aman antar jaringan baik internal maupun eksternal dalam beberapa lapis keamanan dengan fungsi yang berbeda. Pengertian firewall yang lain adalah sistem yang mengimplementasikan aturan keamanan untuk komunikasi antar jaringan komputer.

Bagan keamanan digambarkan sebagai berikut:

Gambar 1. Bagan Keamanan

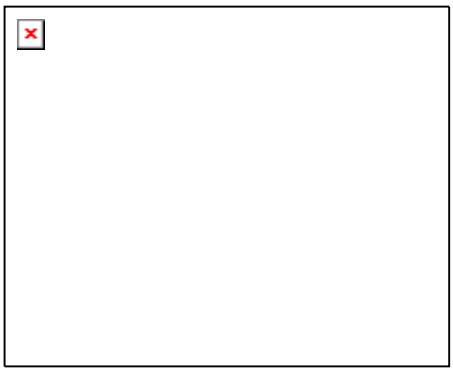


Instalasi dan Konfigurasi pada DMZ

Sampai pada langkah ini maka bagian eksternal jaringan telah terbentuk, dan dapat dikenali sebagai bagian *semi-trusted* (DMZ). Namun dengan catatan

bahwa pada jaringan, koneksi ISDN telah dipisahkan dan dapat berjalan seperti yang diharapkan. Namun biarkan koneksi ADSL offline, hingga pengujian kinerja sistem telah dilakukan. Langkah ini dilakukan untuk meyakinkan bahwa sistem tidak

terganggu selama proses instalasi dan konfigurasi.Sebelumnya arsitektur keamanan yang akan dibangun digambarkan sebagai berikut:



Gambar 2. Arsitektur Keamanan

Langkah yang perlu dilakukan adalah sebagai berikut:

Pertama adalah menentukan subnet yang akan digunakan pada DMZ. 192.100.100.0/14 merupakan tebaran IP yang digunakan pada jaringan. Selain itu dilakukan juga pengujian pada tebaran IP. Tebaran IP harus mempunyai ruang lingkup yang sama.
Ethernet1 pada PIX mempunyai alamat IP 192.100.100.6/24, untuk tetap berhubungan dengan antarmuka Ethernet. Switch juga ditentukan alamat IP-nya. Hal ini dilakukan untuk mencegah hacker memasuki dan melakukan sniffing pada sistem. Langkah ini merupakan proses inspeksi dan antispoofing pada firewall PIX. Setelah langkah diatas dilakukan, maka semua mesin telah terhubung dan online berdasar pada alamat IP dan layanan jaringan komputer yang dibutuhkan. Mesin tersebut dapat dikonfigurasi dengan berbagai spesifikasi keamanan. Firewall PIX juga dapat menerima permintaan VPN yang masuk, untuk memberi kemungkinan bagi pengguna jarak jauh untuk melakukan otentikasi pada sistem. Pembatasan akses pada server VPN menggunakan daftar akses yang menolak semua permintaan koneksi untuk mengakses VPN sampai diverifikasi sebagai salah satu kantor cabang atau dari kantor utama. Akses SSH pada mesin DMZ dapat dikontrol dari komputer lokal dengan menggunakan kantor. Sehingga dimungkinkan mengakses mesin melalui SSH, namun port scan dari internet tidak pernah melihat port SSH terbuka.Langkah berikutnya adalah melakukan redirect layanan DNS dari koneksi ADSL. Translasi alamat port dilakukan pada firewall PIX untuk meneruskan pengiriman setiap permintaan UDP/TCP port 53. Kemudian aktifkan filter paket pada firewall PIX untuk mengizinkan input koneksi TCP port 53 dari NamaServer secondary(umumnya milik ISP), permintaan UDP port 53 yang masuk, dan permintaan UDP port 53 yang keluar dari server DNS. Konfigurasi ini harus diselesaikan untuk semua layanan pada semua server dengan layanan yang munkin berbeda-beda per server. Penelitian yang perlu dilakukan selanjutnya adalah mengubah delegasi nama server sehingga delegasi nameserver yang diharapkan nameserver primer dkonfigurasi untuk alamat eksternal ADSL, bukan untuk ISDN alamat eksternal. Setelah melalui semua langkah diatas, maka koneksi ADSL dapat dikonfigurasi dengan tingkat keamanan yang dibutuhkan oleh semua daftar akses dan PAT. Aturan dasarnya adalah jika tidak membutuhkan akses layanan tertentu, maka akses akan ditolak. Hacker hanya dapat menyerang layanan yang disediakan oleh host DMZ, oleh karena itu upaya yang harus dilakukan adalah meminimalisasi jumlah layanan yang dapat diakses lewat internet, serta melengkapi layanan tersebut dengan keamanan yang tinggi. Sedangkan pada dasarnya semua layanan dapat diakses lewat internet. Lakukan pengujian pada setiap layanan, memindai semua port dan yakinkan

anda mempunyai akses yang terbatas, sebanyak kemungkinan layanan spesifik.

Langkah terakhir adalah melakukan konfigurasi antarmuka ethernet pada firewall PIX ke dalam jaringan internal. Pastikan semua lalu lintas diblok melalui daftar akses dari jaringan internal, sehingga tidak ada orang maupun apapun yang dapat akses keluar.

Konfigurasi Chaining / PassThrough

Saat ini, pada perancangan DMZ membutuhkan proxy untuk semua layanan yang mungkin, sehingga server exchange tidak mengirim dan menerima mail secara langsung, namun dengan proxy semua yang berkaitan dengan mail akan melalui server sendmail. Hal ini berarti bahwa tidak ada pengguna internet yang dapat langsung mengakses melalui layanan internal apapun. Selain itu, kita dapat mengeksekusi multi nilai pada perangkat yang berbeda untuk meningkatkan proteksi. Misalnya dengan mengaktifkan pemindai virus sendmail pada mail server DMZ dan Norton AntiVirus untuk exchange dan servernya.

Daftar akses mempunyai kompleksitas yang bervariasi tergantung pada layanan yang diaktifkan. Misalnya permintaan untuk menyediakan informasi web untuk kepentingan umum, membutuhkan persetujuan dari Proxy pada DMZ hanya dari proxy internal, dan Proxy DMZ dan menginisilisasi semua koneksi ke luar. Proxy DMX tidak perlu menerima request persetujuan dari sumber yang lain, sehingga hacker seharusnya tidak dapat mengakses server.

Keamanan data merupakan isu yang menjadi titik tolak pembahasan makalah ini, oleh karena itu, perlu dilakukan suatu langkah yang dapat memastikan lebih lanjut bahwa tidak terdapat pihak yang tidak memerlukan data, mempunyai hak akses. Misalnya permintaan akses web hanya diperuntukan bagi lingkungan internalh dengan mekanisme proxy dan proxy DMZ. Pada koneksi SMTP, permintaan layanan DNS seharusnya datang dari server DNS pada jaringan internal, dan seharusnya hanya dimiliki oleh DMZ named server. Segala sesuatu yang tidak diperlukan, sebaiknya secara otomatis di blok. Lakukan batasan sebanyak mungkin.

Sebagian besar sistem, hanya melakukan pembatasan layanan hanya untuk permintaan dari pihak eksternal, namun sebaiknya pembatasan layanan juga diberikan bagi permintaan dari internal jaringan. Hal ini disebabkan oleh dua faktor. Pertama, penyederhanaan file log. Beberapa trojan dikonfigurasi dengan melakukan pemanggilan inisialisasi dari mesin yang terinfeksi menuju mesin lain yang akan diserang. Oleh karena itu jika administrator teledor dan tidak memberi batasan pada permintaan keluar, maka hal ini akan memberikan peluang bagi hacker untuk memasuki sistem, terlepas dari tingkat keamanan yang terdapat dari sisi penerima. Selain itu hilangkan kemungkinan pada setiap mesin untuk dapat melakukan permintaan layanan pada diri sendiri(*loop request*). Jika dibutuhkan melakukan IRC ke internet, gunakan server IRC. Namun jangan memberi layanan ICQ atau MSN secara internal. Pada dasarnya sampai saat ini, telah dilakukan pengujian pada tingkat keyakinan koneksi keluar dari jaringan internal, yang hanya dapat dilakukan dari server.

Langkah 6 – Alarm dan Tripwire

Terdapat banyak metode untuk mencegah hacker, namun masih terlalu banyak wilayah komputerisasi yang dapat dijelajahi. Sehingga, adu

kekuatan antara administrator dan hacker tidak akan reda dalam waktu singkat, bahkan diperkirakan akan terus berjalan seiring dengan perkembangan teknologi. Pada makalah ini akan dibahas tiga metode utama, yaitu:

SysLog Logging

Pendeteksi serangan Tripwire

Cron Jobs

Pada firewall PIX, log semua kesalahan otorisasi dan koneksi ke server SysLog pada jaringan internal. Sehingga dengan log yang disimpan, dapat diketahui bagaimana metode yang digunakan hacker untuk memasuki sistem.

Tripwire merupakan daemon yang baik bagi sistem dan dapat memonitor kumpulan file yang diubah atau dimodifikasi. Jika hacker mencoba memasuki sistem, maka apapun yang dilakukan akan selalu memodifikasi file. Sehingga administrator dapat memonitor setiap perubahan pada file, dan dapat menentukan kompromi yang diambil. Catat aktifitas tersebut pada server SysLog, atau bahkan eksekusi script yang bersesuaian dengan file tersebut.

Tripwire mempunyai satu kelemahan. Proses tripwire bekerja pada sistem dengan interval tertentu, dan para hacker telah mengetahui kelemahan ini, dan jika mereka telah memperoleh hak akses root, maka mereka akan menonaktifkan proses. Oleh karena itu perlu dibangun sebuah script (*cron job*) yang selalu mencari pada proses yang sedang aktif untuk memastikan tripwire sedang aktif. Jika diketahui tripwire telah dinonaktifkan maka script akan mengirimkan email dan sms ke administrator dan mesin di*shutdown*. Proses ini pun mempunyai resiko data dan program menjadi rusak. Namun jika terdapat mail exchanger ISP cadangan, dan server DNS cadangan, maka sebelum di*shutdown*, program dan data diselamatkan ke server cadangan.

Pendeteksi Serangan

Metode Pendeteksi Serangan bekerja dengan asumsi bahwa aktivitas penyerang dapat dibedakan dengan aktivitas yang dilakukan pengguna pada umumnya. Perbedaan karakteristik antara pengguna dan penyerang termasuk pada kumpulan parameter yang dikaji dan awal data yang berkomunikasi.

Pendeteksi serangan *Host-Based* terdiri dari dua tipe yaitu spesifik pada aplikasi dan spesifik pada sistem operasi. Pada kedua tipe tersebut agen umumnya berada pada server yang selanjutnya akan dimonitor, melakukan analisis file log, pengaksesan data, dan file log aplikasi. Sistem keamanan dengan host-based menggunakan teknik perbandingan antara modul yang dideteksi mengalami anomali dengan pola normal secara statistik. Pada kasus monitor untuk tipe spesifik pada sistem operasi, sesi yang abnormal seperti login yang tidak berhasil akan dibandingkan dengan model behavioral dari penggunaan normal dengan menggunakan kriteria seperti waktu akses, serta jumlah dan tipe file yang diakses atau dibuat. Pendeteksi serangan spesifik pada aplikasi umumnya mendefinisikan sekumpulan aturan yang menggambarkan aktivitas dengan acuan kejadian log.

Pendeteksi Serangan di Masa Depan

Seakan setiap waktu, dibangun dan beredar perangkat lunak baru, namun masa depan adalah milik penganalisis syslog. Perangkat bantu syslog menyediakan informasi syslog, memprosesnya, dan jika kondisi tertentu dipenuhi, maka akan mengaktifkan script. Umumnya para

hacker memulai aktivitasnya dengan melakukan pemindaian port. Administrator dapat mengkonfigurasi firewall pada umumnya, untuk mencatat(log) semua kesalahan komunikasi pada server syslog.
Jika terdapat penganalisis, maka dapat dibentuk kumpulan aturan yang akan mengeksekusi script jika terdapat penolakan 4 koneksi pada port atau protokol yang berbeda dengan periode 10 detik. Script tersebut secara otomatis menambah pernyataan *deny* pada daftar akses firewall, yang selanjutnya firewall akan menolak semua koneksi dari alamat IP dari port diluar sistem. Daftar *deny* tersebut dapat disimpan dalam waktu 30 menit, namun berdampak pada seksi pengumpulan informasi. Dengan kemampuan fungsi tersebut, maka administrator dapat mengendalikan tingkat keamanan sistem sebagaimana yang dibutuhkan.

Langkah 7 – Aktifkan Sistem

Setelah melalui langkah-langkah diatas, maka sistem dapat diaktifkan, dengan tetap selalu mencatat perubahan-perubahan yang terjadi. Hal lain yang diperlukan sistem adalah ubah nama record, konfigurasi ulang layanan internal dan siap untuk digunakan.

Pemrograman Jaringan pada Protokol TCP/IP

Protokol TCP/IP merupakan protokol yang banyak digunakan pada lingkungan internal jaringan, bahkan merupakan protokol standard yang digunakan pada komunikasi internet. Berkaitan dengan pembangunan sistem keamanan, maka selain melakukan pencegahan dengan DMZ, diperlukan pula pengetahuan teknik pemrograman jaringan (*Network Programming*), sehingga dengan semikian diharapkan administrator lebih mengenal bagaimana cara kerja hacker dan kelompok penyerang yang lain.
Network Programming menggunakan bahasa C++ untuk beberapa metoda akses jaringan komunikasi data, yaitu:
Mengetahui nama sebuah komputer dan alamat IP-nya
Pendeteksian, dan penutupan koneksi pada TCP/IP
Mengetahui nama komputer lain dan alamat IP masing-masing
Pendeteksian port pada TCP/IP
Melakukan operasi ping pada TCP/IP
Pendeteksian alamat MAC

Mengetahui Nama Sebuah Komputer dan Alamat IP-nya

Kode program dan langkah-langkah berikut dapat digunakan untuk mengetahui nama komputer dan alamat IP pada komputer yang mengeksekusi program tersebut.

```
#include <winsock2.h>

{
   WORD wVersionRequested;
   WSADATA wsaData;
   char name[255];
   CString ip;
   PHOSTENT hostinfo;
   wVersionRequested          =
MAKEWORD( 2, 0 );

   if       (       WSAStartup(
wVersionRequested,  &wsaData  )
== 0 )
   {
      if( gethostname ( name,
sizeof(name)) == 0)
      {
         if((hostinfo          =
gethostbyname(name)) != NULL)
            {

ip=inet_ntoa(*(struct    in_addr
*)*hostinfo->h_addr_list);
            }
      }

      WSACleanup( );
   }
}
```

Pendeteksian, dan Penutupan Koneksi Pada TCP/IP

Pada saat salah satu terminal yang berkoneksi dengan TCP/IP terminate atau down tanpa mengirimkan sinyal ke terminal lain, dapat mengakibatnya terminal lain yang juga sedang berkoneksi dalam kondisi deadlock. Tugas utama server adalah mengelola sejumlah besar koneksi. Jika terdapat satu socket yang hang pada server, dapat mengakibatkan pemborosan resource. Winsock tidak dapat menyediakan layanan yang dapat melalu memantau adanya terminal yang terminate/down pada koneksi TCP/IP. Sehingga dibutuhkan fungsi HasConnectionDropped yang dapat melakukan fungsi tersebut.

Langkah-langkah pada fungsi tersebut adalah:

1. Cek apakah socket dapat dibaca
2. Jika Ya, deteksi data yang masuk
3. Cek nilai data dan error untuk menentukan apakah koneksi jaringan masih terjaga.

Pendeteksian data yang datang dilakukan dengan flag MSG_PEEK. Kemudian data disalin pada buffer tanpa menghapus data dari antrian input, sehingga tidak mengganggu kerja sistem. Pendeteksian error dilakukan jika recv call mengembalikan nilai error. Jika recv call mengembalikan nilai 0, maka dapat disimpulkan adanya permintaan pemutusan hubungan dari terminal yang lain. Nilai ini dapat juga digunakan untuk mendeteksi kegagalan koneksi.

Berikut adalah kode fungsi yang telah dimodifikasi:

```
BOOL
CClientSocket::HasConnectionDro
pped( void )
{
        BOOL    bConnDropped    =
FALSE;
        INT iRet = 0;
        BOOL bOK = TRUE;

        struct  timeval  timeout
= { 0, 0 };
        fd_set readSocketSet;

        FD_ZERO(  &readSocketSet
);
        FD_SET(       m_hSocket,
&readSocketSet );

        iRet   =   ::select(   0,
&readSocketSet,    NULL,    NULL,
&timeout );
        bOK = ( iRet > 0 );

        if( bOK )
        {
                bOK  =  FD_ISSET(
m_hSocket, &readSocketSet );
        }

        if( bOK )
        {
                CHAR szBuffer[1]
= "";
                iRet   =   ::recv(
m_hSocket,      szBuffer,      1,
MSG_PEEK );
                bOK = ( iRet > 0
);
                if( !bOK )
                {
                        INT
iError = ::WSAGetLastErrrr();

        bConnDropped    =    (    (
iError == WSAENETRESET ) ||
                                (
iError == WSAECONNABORTED ) ||
                                (
iError == WSAECONNRESET ) ||
                                (
iError == WSAEINVAL ) ||
                                (
iRet == 0 ) );
                }
        }

    return( bConnDropped );
}
```

Mengetahui Informasi Mengenai Workstation

Program berikut digunakan untuk mengetahui informasi mengenai workstation baik yang aktif maupun yang tidak, tanpa mengubah konfigurasi server. Fasilitas ini cocok digunakan untuk aplikasi monitoring jaringan secara real time.

```
#include <what_you_need.h>
#include <lmcons.h>
#include <lmwksta.h>
#include <lmserver.h>
#include <lmerr.h>*/

//      Network   API   job   -
obtain   network   info   about
selected machine.
BOOL
_GetWkstaInformation100()
{
        LPBYTE lpBuf;
        LPCSTR   lpcstrWkstaName
= (LPCSTR)m_strWkstaName;
        int   iwLength   =   2   *
(MAX_COMPUTERNAME_LENGTH + 1);
        WCHAR  lpwWkstaName[2  *
(MAX_COMPUTERNAME_LENGTH + 1)];
        lpwWkstaName[0] = '\0';
        MultiByteToWideChar(CP_
ACP,  0,  lpcstrWkstaName,  -1,
lpwWkstaName, iwLength);

        typedef   NET_API_STATUS
(NET_API_FUNCTION
*NETWKPROC)(LPWSTR,       DWORD,
LPBYTE *);

        NETWKPROC
_procNetWkstaGetInfo          =
(NETWKPROC)

        (GetProcAddress(theApp.
m_hNetDLL,
_T("NetWkstaGetInfo")));
        if(_procNetWkstaGetInfo
)
        {
                NET_API_STATUS
nasRetVal                     =
(*_procNetWkstaGetInfo)(lpwWkst
aName, 100, (LPBYTE*)&lpBuf);
```

```
                if(nasRetVal   ==
NERR_Success)
                {

        WKSTA_INFO_100
*pWkstaInfo   =   (WKSTA_INFO_100
*)lpBuf;
                        DWORD
dwPlatformId    =    pWkstaInfo-
>wki100_platform_id;


        if(dwPlatformId       !=
PLATFORM_ID_NT)
                        {

        //[ERROR]Not  a  Windows
NT Workstation - if useful.

        return FALSE;
                        }
                        else
                          return
TRUE;
                }
                else
                {

        //[ERROR]  System  error.
Call              GetLastError,
FormatMessage, etc.
                        return
FALSE;
                }
        }
        else
        {
                //[ERROR]Unable
to        find        procedure
NetWkstaGetInfo               in
netapi32.dll.
                return FALSE;
        }
}
```

Mengetahui nama komputer lain dan alamat IP masing-masing

Program ini dapat digunakan untuk mendapatkan informasi mengenai terminal-terminal yang berkoneksi dengan jaringan TCP/IP. Fungsi ini sama dengan fungsi Network Neighbourhood pada MsWindows.

Langkah-langkah yang dilakukan adalah sebagai berikut:

1. Include winsock2.h
2. Pada Menu, pilih Project-Setting dan pada tab Link, pilih Object/Library Modules
3. Tambahkan ws2_32.lib dan mpr.lib pada daftar link sebelumnya
4. Kompilasi kode program berikut tanpa membuat linker error

```
CString strTemp;
struct hostent *host;
struct in_addr *ptr;   //     To
retrieve the IP Address

DWORD          dwScope          =
RESOURCE_CONTEXT;
NETRESOURCE    *NetResource     =
NULL;
HANDLE hEnum;
WNetOpenEnum(   dwScope,   NULL,
NULL, NULL, &hEnum );

WSADATA wsaData;
WSAStartup(MAKEWORD(1,1),&wsaDa
ta);

if ( hEnum )
{
  DWORD Count = 0xFFFFFFFF;
  DWORD BufferSize = 2048;
  LPVOID     Buffer     =     new
char[2048];
  WNetEnumResource(       hEnum,
&Count, Buffer, &BufferSize );
  NetResource                  =
(NETRESOURCE*)Buffer;

  char szHostName[200];

  for ( unsigned int i = 0; i <
BufferSize/sizeof(NETRESOURCE);

i++, NetResource++ )
  {
    if   (   NetResource->dwUsage
== RESOURCEUSAGE_CONTAINER &&
                  NetResource-
>dwType == RESOURCETYPE_ANY )
    {
      if      (      NetResource-
>lpRemoteName )
      {
        CString strFullName =
NetResource->lpRemoteName;
        if      (      0      ==
strFullName.Left(2).Compare("\\
\\") )
          strFullName         =
strFullName.Right(strFullName.G
etLength()-2);
        gethostname(
szHostName, strlen( szHostName
) );
        host                   =
gethostbyname(strFullName);
        if(host     ==     NULL)
continue;
        ptr = (struct in_addr
*) host->h_addr_list[0];

        //   Eg.   211.40.35.76
split up like this.
        int     a     =     ptr-
>S_un.S_un_b.s_b1;  // 211
        int     b     =     ptr-
>S_un.S_un_b.s_b2;  // 40
        int     c     =     ptr-
>S_un.S_un_b.s_b3;  // 35
        int     d     =     ptr-
>S_un.S_un_b.s_b4;  // 76

        strTemp.Format("%s  -->
%d.%d.%d.%d",strFullName,a,b,c,
d);

AfxMessageBox(strTemp);
      }
    }
  }
  delete Buffer;
  WNetCloseEnum( hEnum );
}

WSACleanup();
```

Pendeteksian Port Pada TCP/IP

Program berikut merupakan pendeteksi port yang efektif yang dilengkapi dengan fasilitas pengujian koneksi dan layanan basis data lokal. Pengetahuan dasar yang dibutuhkan adalah MFC dan Winsock API.

Mekanisme kerja program berikut adalah, setiap terjadi pengiriman atau penerimaan data

melalui internet, maka aplikasi e-mail harus berkoneksi dengan remote port pada remote terminal. Beberapa layanan yang tersedia pada internet adalah sebagai berikut:

Layanan	Port	Deskripsi
echo	7	Echo
daytime	13	Jam
ftp	21	File Transfer Protocol
Ssh	22	SSH Remote Login Protocol
telnet	23	Telnet
smtp	25	Simple Mail Transfer
time	37	Waktu
nameserver	42	Host Name Server
nicname	43	Who Is
domain	53	Domain Name Server
gopher	70	Gopher
http	80	World Wide Web HTTP
kerberos	88	Kerberos
pop3	110	Post Office Protocol
netbios-ns	137	NETBIOS Name Service
netbios-dgm	138	NETBIOS Datagram Service
netbios-ssn	139	NETBIOS Session Service

Namun sebenarnya layanan atau port yang tersedia sangat banyak, dan dapat dikelompokkan sebagai berikut:

Port	Fungsi
0 – 1023	Port yang sering digunakan pada layanan server, seperti smtp, pop3, ftp, etc.
1024 - 49151	Port yang terdaftar pada organisasi IANA
49152 - 65535	Dinamik dan/atau private port yang bebas digunakan

Berdasarkan RFC793 mengenai TCP dan RFC768 mengenai protokol UDP, port-port diatas dapat digunakan pada koneksi TCD dan UDP. Sehingga tidak muncul duplikasi entri yang menggunkaan layanan yang sama, seperti berikut:

Service	Port/Protocol	Deskripsi
echo	7/tcp	Echo
echo	7/udp	Echo

Protokol TCP/IP mengacu pada model OSI (Open Systems Interconnection) yang diperkenalkan oleh ISO (International Organization for Standardization). Model OSI terdiri dari 7 lapis yaitu Aplikasi, Presentasi, Sesi, Transport, Network, Data Link, dan Fisik. Pada lapis bawah terdapat NIC yang digunakan untuk komunikasi data antar komputer. TCP/IP terdiri dari protokol-protokol:

Protokol	Deskripsi
IP (Internet Protocol)	Protokol yang digunakan untuk komunikasi data antar terminal
TCP (Transport Control Layer)	Protokol yang digunakan untuk komunikasi data antar aplikasi

UDP (User Data Protocol)	Digunakan oleh TCP untuk komunikasi data antar aplikasi dengan mekanisme yang lebih sederhana dan lebih reliable daripada TCP
ICMP (Internet Control Message Protocol)	Digunakan pada lapis Network untuk error messaging

Hubungan antara model OSI dan ptotokol TCP/IP dapat dituliskan sebagai berikut:

Model OSI	TCP/IP
application layer	client program
presentation layer	client program
session layer	client program
transport layer	TCP/UDP
network layer	ICMP/IP/IGMP
data-link layer	ARP/hardware/RARP
Phisical layer	ARP/hardware/RARP

Kode program berikut merupakan WinSock programming dalam mode asynchronous bukan blocking atau non blocking. Program akan memonitor 1 port dalam 1 waktu tanpa terjadi blocking hingga semua port dicek.

Kode program dibagi menjadi 2 bagian yaitu CPropertySheet dalam bentuk aplikasi MFC dan class untuk antarmuka WinSock.

File CPropertySheetDialog.cpp/.h dan CPropertyPageDialog.cpp/.h terdiri dari semua class yang dibutuhkan pada mekanisme Property Sheet dan digunakan sebagai basis aplikasi MFC. File TcpPropertySheet.cpp/.h terdiri dari kade untuk aplikasi utama, sedangkan semua file Tcp[...]Page.cpp/.h terdiri dari kode untuk halaman sheet. Kode antarmuka WinSock API terdapat pada file CWinsock.cpp/.h dan CSock.cpp/.h. File CAsyncSock.cpp/.h terdiri dari class yang digunakan untuk mengakses Winsock API dalam mode asyncronous, sedangkan kode program yang lain digunakan secara internal untuk menangani daftar kontrol, konfigurasi program, dan sebagainya.

Class pembangun antarmuka Winsock adalah CWinsock yang berfungsi untuk memetakan semua layanan WinSock ke library atau dummy inplementation tergantung pada definisi makro _DEBUGSOCKET, karena class CwinSock terdiri dari kode program untuk menangani layanan minimal server SMTP/POP3, yang dapat digunakan untuk menguji protokol SMTP/POP3 tanpa membutuhkan koneksi internet.

Umumnya aplikasi MFC membutuhkan class CTcpScanApp dari CwinApp yang didefinisikan sendiri dengan mengubah InitInstance() pada implementasi property sheet dan semua page yang berkaitan.

```
BOOL
CTcpScanApp::InitInstance(void)
{
  CScanPage ScanPage;
  CConnectPage ConnectPage;
  CServicesPage ServicesPage;
  CPropertyPageList*
pPropertyPageList     =     new
CPropertyPageList();

  if(pPropertyPageList)
  {
```

```
     PROPERTYPAGE* p;

     p           =           new
PROPERTYPAGE(IDD_PAGE_SCAN,

&ScanPage,

RUNTIME_CLASS(CScanPage));
     pPropertyPageList->Add(p);

     p           =           new
PROPERTYPAGE(IDD_PAGE_CONNECT,

&ConnectPage,

RUNTIME_CLASS(CConnectPage));
     pPropertyPageList->Add(p);

     p           =           new
PROPERTYPAGE(IDD_PAGE_SERVICES,

&ServicesPage,

RUNTIME_CLASS(CServicesPage));
     pPropertyPageList->Add(p);

     CTcpScanPropertySheet*
pPropertySheetDialog =
              new
CTcpScanPropertySheet(NULL,

pPropertyPageList);
     if(pPropertySheetDialog)
     {
       if(pPropertySheetDialog-
>Create())
       {
          m_pMainWnd          =
pPropertySheetDialog;
          pPropertySheetDialog-
>DoModal();
       }

       delete
pPropertySheetDialog;
    }

    delete pPropertyPageList;
  }

  return(FALSE);
}
```

Dialog property sheet merupakan basis class CpropertySheetDialog / CPropertyPageDialog, yang menangani semua hal yang dibutuhkan. Button *Apply* danthe *Help* dihapus sehingga hanya tersisa button *OK* dan *Cancel*.

Setiap class yang digunakan sebagai page pada sheet didefinisikan oleh handle berikut:

```
BOOL  OnInitDialog(void);
BOOL  OnSetActive(void);
BOOL  OnKillActive(void);
void  OnKillSheet(void);
void  OnOk(void);
void  OnCancel(void);
```

Melakukan operasi ping pada TCP/IP

Masalah umum pada TCP/IP adalah bagaimana melakukan ping pada Windows dengan menggunakan stack MS-TCP. Masalah tersebut dapat ditangani dengan menggunakan ICMP DLL.

Masalah implementasi adalah jika ditentukan nama komputer, atau alamat IP, maka lakukan ping yang dapat mengembalikan informasi waktu response ping. Fungsi ini membutuhkan ICMP.DLL dan beberapa struktur socket pada Csocket. Sebelum melakukan percobaan, file ICMPAPI.H, ICMP.LIB, dan IPEXPORT.H dari Microsoft, diletakkan pada direktori lib.

Class terdiri dari 4 fungsi publik:

```
unsigned long ResolveIP(CString
strIP)
unsigned                    long
ResolveName(CString
strHostName)
CString   GetIP(unsigned   long
ulIP)
DWORD   PingHost(unsigned   long
ulIP, int iPingTimeout)
```

ResolveIP mengambil informasi alamat IP CString (seperti "123.123.123.123"), dan

mengembalikan alamat network berdasarkan urutan byte.

ResolveName mengambil inormsi host name CString, yang diakses melalui DNS atauWINS, dan mengembalikan alamat network berdasarkan urutan byte.

GetIP mengambil informasi alamat network berdasarkan urutan byte, dan mengembalikan alamat IP sebagai Cstring.

PingHost mengambil informasi alamat network berdasarkan urutan byte, timeout integer, alamat ping, and mengembalikan ping response time.

```
/*
//--------------------------------
--------------------------------
--------------------------------
-----------------------
//icmpecho.h
//--------------------------------
--------------------------------
--------------------------------
-----------------------
*

class CIcmpEcho : public
CSocket
{
// Attributes
public:

// Operations
public:
        CIcmpEcho();
        virtual ~CIcmpEcho();

        unsigned            long
ResolveIP(CString strIP);
        unsigned            long
ResolveName(CString
strHostName);

        DWORD  PingHost(unsigned
long ulIP, int iPingTimeout);

        CString   GetIP(unsigned
long ulIP);

// Overrides
public:
        //           ClassWizard
generated   virtual   function
overrides
        //{{AFX_VIRTUAL(CIcmpEc
ho)
        //}}AFX_VIRTUAL

        //   Generated   message
map functions
        //{{AFX_MSG(CIcmpEcho)
                //  NOTE  -  the
ClassWizard will add and remove
member functions here.
        //}}AFX_MSG

// Implementation
protected:
};
/////////////////////////////////
/////////////////////////////////
///////////////


/*
//--------------------------------
--------------------------------
--------------------------------
-----------------------
//icmpecho.cpp
//--------------------------------
--------------------------------
--------------------------------
-----------------------
*

//        IcmpEcho.cpp        :
implementation file
//

#include "IcmpEcho.h"

extern "C" {
#include "ipexport.h"
#include "icmpapi.h"
}

#define PING_TIMEOUT 100

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static   char   THIS_FILE[]   =
__FILE__;
#endif

/////////////////////////////////
/////////////////////////////////
///////////////
// CIcmpEcho

CIcmpEcho::CIcmpEcho()
```

```
{
}

CIcmpEcho::~CIcmpEcho()
{
}


// Do not edit the following lines, which are needed by ClassWizard.
#if 0
BEGIN_MESSAGE_MAP(CIcmpEcho, CSocket)
        //{{AFX_MSG_MAP(CIcmpEcho)
        //}}AFX_MSG_MAP
END_MESSAGE_MAP()
#endif  // 0

////////////////////////////////////////////////////////////////
// CIcmpEcho member functions
unsigned long CIcmpEcho::ResolveIP(CString strIP)
{
        //Task 1:       Given IP Address i.e. "111.111.111.111",
        //      Return  Network byte ordered address (ulIP)

        unsigned long ulIP;

        ulIP =(IPAddr)inet_addr(strIP);

        return ulIP;
}

unsigned long CIcmpEcho::ResolveName(CString strHostName)
{
        //Task 1:       Resolve HostName (through DNS or WINS, whichever appropriate)
        //Task 2:       Return network byte ordered address (ulIP)

        unsigned long ulIP;
        hostent* phostent;

        phostent = gethostbyname(strHostName);

        if (phostent == NULL)
                return 0;

        ulIP = *(DWORD*)(*phostent->h_addr_list);

        return ulIP;

}

DWORD CIcmpEcho::PingHost(unsigned long ulIP, int iPingTimeout)
{
        //Task 1:       Open ICMP Handle
        //Task 2:       Create Structure to receive ping reply
        //Task 3:       SendPing (SendEcho)
        //Task 4:       Close ICMP Handle
        //Task 5:       Return RoundTripTime

        unsigned long ip = ulIP;

        HANDLE icmphandle = IcmpCreateFile();

        char reply[sizeof(icmp_echo_reply)+8];

        icmp_echo_reply* iep=(icmp_echo_reply*)&reply;
        iep->RoundTripTime = 0xffffffff;

        IcmpSendEcho(icmphandle,ip,0,0,NULL,reply,sizeof(icmp_echo_reply)+8,iPingTimeout);

        IcmpCloseHandle(icmphandle);

        return iep->RoundTripTime;

}

CString CIcmpEcho::GetIP(unsigned long ulIP)
{
        //Task 1:       Given a host order ulIP Address
        //      Return  a IP address in format of xxx.xxx.xxx.xxx

        LPSTR szAddr;
```

```
        struct            in_addr
inetAddr;

        inetAddr.s_addr         =
(IPAddr)ulIP;

        szAddr                  =
inet_ntoa(inetAddr);

        CString csIP = szAddr;

        return csIP;

}
```

Pendeteksian alamat MAC

Alamat MAC dapat dilakukan dengan berbagai cara, salah satunya adalah dengan melakukan queri driver miniport NDIS. Mekanisme miniport dapat dibagi menjadi beberapa metode yaitu:

Metode 1: UuidCreate

Cara paling sederhana untuk mengetahui alamat MAC, yaitu dengan membuat Uuid sekuensial. Microsoft menggunakan alamat MAC untuk membangun universally unique identifier.

Langkah yang dilakukan adalah mencek byte ke-2 hingga ke-8. Kode program sebagai berikut:

```
// Fetches the MAC address and
prints it
static void GetMACaddress(void)
{
  unsigned char MACData[6];

  UUID uuid;
  UuidCreateSequential(    &uuid
);    // Ask OS to create UUID
  for  (int  i=2;  i<8;  i++)   //
Bytes 2 through 7 inclusive
                              //
are MAC address
    MACData[i     -     2]     =
uuid.Data4[i];
  PrintMACaddress(MACData);
// Print MAC address
}
```

Metode ini hanya dapat dieksekusi pada PC dengan NIC tunggal.

Metode 2: Menggunakan NetBIOS

Metode ini mendukung PC dengan multi NIC, namun membutuhkan NetBIOS yang harus di install beserta koneksi kabel yang valid untuk NetBIOS.

```
// Fetches the MAC address and
prints it
static void GetMACaddress(void)
{
  unsigned   char   MACData[8];
// Allocate data structure

// for MAC (6 bytes needed)

  WKSTA_TRANSPORT_INFO_0
*pwkti;   //   Allocate   data
structure

// for NetBIOS
  DWORD dwEntriesRead;
  DWORD dwTotalEntries;
  BYTE *pbBuffer;

  //   Get   MAC   address   via
NetBIOS's enumerate function
  NET_API_STATUS   dwStatus   =
NetWkstaTransportEnum(
   NULL,                      //
[in]  server name
   0,                         //
[in]  data structure to return
   &pbBuffer,                 //
[out] pointer to buffer
   MAX_PREFERRED_LENGTH,      //
[in]  maximum length
   &dwEntriesRead,            //
[out] counter of elements
                              //
actually enumerated
   &dwTotalEntries,           //
[out] total number of elements
                              //
that could be enumerated
   NULL);                     //
[in/out] resume handle
```

```
  assert(dwStatus              ==
NERR_Success);

  pwkti                         =
(WKSTA_TRANSPORT_INFO_0
*)pbBuffer;  //  type  cast  the
buffer

  for(DWORD       i=1;       i<
dwEntriesRead;  i++)    //  first
address is

// 00000000, skip it
  {
// enumerate MACs & print
    swscanf((wchar_t
*)pwkti[i].wkti0_transport_addr
ess,

L"%2hx%2hx%2hx%2hx%2hx%2hx",
            &MACData[0],
            &MACData[1],
            &MACData[2],
            &MACData[3],
            &MACData[4],
            &MACData[5]);
    PrintMACaddress(MACData);
  }
  // Release pbBuffer allocated
by above function
  dwStatus                      =
NetApiBufferFree(pbBuffer);
  assert(dwStatus              ==
NERR_Success);
}
```

Metode 3: Menggunakan GetAdaptersInfo

Cara yang paling banyak dilakukan untuk mendapatkan informasi alamat MAC adalah dengan metode GetAdaptersInfo. Metode tersebut dapat menyediakan banyak informasi sebagai IPCONFIG /ALL termasuk server DHCP, Gateway, daftar alamat IP, subnet mask, dan WINS server.

```
// Fetches the MAC address and
prints it
static void GetMACaddress(void)
{
  IP_ADAPTER_INFO
AdapterInfo[16];              //
Allocate information

// for up to 16 NICs
  DWORD       dwBufLen        =
sizeof(AdapterInfo);    //  Save
memory size of buffer

  DWORD       dwStatus        =
GetAdaptersInfo(        // Call
GetAdapterInfo
    AdapterInfo,
// [out] buffer to receive data
    &dwBufLen);
//  [in]  size  of  receive  data
buffer
  assert(dwStatus              ==
ERROR_SUCCESS);      //   Verify
return value is

// valid, no buffer overflow

  PIP_ADAPTER_INFO  pAdapterInfo
=   AdapterInfo;   //   Contains
pointer to

// current adapter info
  do {

PrintMACaddress(pAdapterInfo-
>Address); // Print MAC address
    pAdapterInfo                =
pAdapterInfo->Next;           //
Progress through

// linked list
  }
  while(pAdapterInfo);
// Terminate if last adapter
}
```

Kesimpulan

Untuk mencapai suatu keamanan suatu jaringan komputer yang optimal diperlukan suatu koordinasi antara pengguna dan Administrator serta aturan/rule atau otorisasi dalam penggunaan jaringan tsb . Dilain pihak agar diupayakan adanya update software secara berkala serta menyesuaikan hardwarenya dengan perkembangan teknologi terbaru.

Daftar Pustaka

http://compnetworking.about.com/cs/networksecurity/g/bldef_dmz.htm

Intrusion Detection within a Secured Network, Secure System Administrating Research, 1999

Marek, Building Secure Network with DMZ's, 2002

Spitzner, Lance, A Passive Approach to Your Network Security, "The Secrets of Snoop"

www.cert.org

Zuliansyah, Mochammad, Teknik Pemrograman Network Interface Card pada Protokol TCP/IP, ITB, 2002