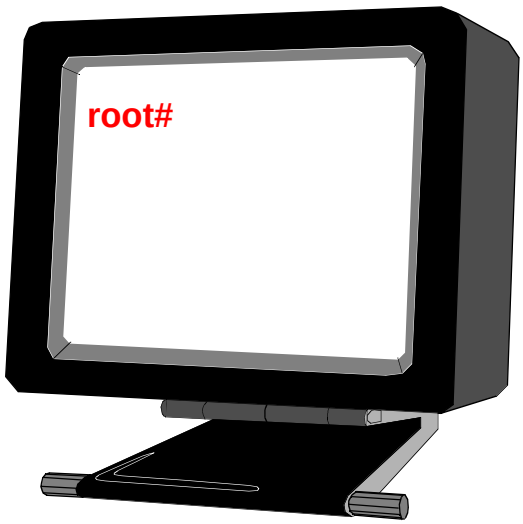
Keamanan Sistem Informasi Berbasis Internet

Budi Rahardjo



PT Insan Komunikasi / Infonesia - Bandung
1998, 1999

Keamanan Sistem Informasi Berbasis Internet

Budi Rahardjo

Distribution and Printing History:

Versi 1.0 mulai didistribusikan secara gratis di Internet dengan format Adobe PDF (Juli 1998). Salah satu tujuan penerbitan gratis ini adalah agar masalah keamanan sistem informasi dapat dimengerti oleh para profesional Indonesia. Penulis berhak mencabut kembali distribusi ini apabila diperlukan. Umpan balik, koreksi, donasi, dan lain-lain harap diarahkan kepada penulis melalui media elektronik.
Total halaman: 45 halaman isi, 5 halaman cover (50 halaman)
E-mail: <rahardjo@Insan.Co.Id>

Versi 2.0. Terima kasih kepada beberapa pembaca yang telah memberikan umpan balik dan koreksi, antara lain dari IECI, Irvan Nasrun, dan masih banyak lainnya yang tidak dapat saya sebutkan satu persatu. Bagian yang diperbaiki antara lain:
Bab "Mengamankan Sistem Informasi" mendapat tambahan yang cukup signifikan.
Adanya daftar indeks.
Total: 54 halaman isi.

Versi 3.0. Penambahan Bab "Keamanan Sistem WWW" dan Bab "Eksploitasi Lubang Keamanan".

Versi 3.1. Terima kasih kepada Indra Dermawan dan Saptoto Aji (mahasiswa Teknik Elektro ITB) yang telah menyumbangkan informasi tambahan tentang serangan terhadap sistem keamanan. Tambahan lain berupa keterangan tentang DES. Beberapa materi dari buku ini sudah diuji dalam Short Course Implementing Security yang diadakan oleh PIKSI ITB.
Jumlah halaman: 64 (total)

Versi 3.2. Tambah informasi tentang hackers, crackers, dan etika. Digunakan untuk materi kuliah EL 776 di Pasca Sarjana, Institut Teknologi Bandung, tahun 1999.
Jumlah halaman: 76 (total)

Versi 3.3. Menambahkan beberapa informasi di berbagai bab, antara lain: queso, nmap, smurf, ntop.
Jumlah halaman: 80 (total), 16 Mei 1999.

ISBN 0-000-000000-0
ABCDEFGHIJ-DO-89

BAB 1

Pendahuluan

Computer Security is preventing attackers from achieving objectives through unauthorized access or unauthorized use of computers and networks.
(John D. Howard, "An Analysis Of Security Incidents On The Internet 1989 - 1995")

Masalah keamanan merupakan salah satu aspek penting dari sebuah sistem informasi. Sayang sekali masalah keamanan ini sering kali kurang mendapat perhatian dari para pemilik dan pengelola sistem informasi. Seringkali masalah keamanan berada di urutan kedua, atau bahkan di urutan terakhir dalam daftar hal-hal yang dianggap penting. Apabila menggangu performansi dari sistem, seringkali keamanan dikurangi atau ditiadakan [6]. Buku ini diharapkan dapat memberikan gambaran dan informasi menyeluruh tentang keamanan sistem informasi dan dapat membantu para pemilik dan pengelola sistem informasi dalam mengamankan informasinya.

Informasi saat ini sudah menjadi sebuah komoditi yang sangat penting. Bahkan ada yang mengatakan bahwa kita sudah berada di sebuah "*information-based society*". Kemampuan untuk mengakses dan menyediakan informasi secara cepat dan akurat menjadi sangat esensial bagi sebuah organisasi, baik yang berupa organisasi

komersial (perusahaan), perguruan tinggi, lembaga pemerintahan, maupun individual (pribadi). Hal ini dimungkinkan dengan perkembangan pesat di bidang teknologi komputer dan telekomunikasi. Dahulu, jumlah komputer sangat terbatas dan belum digunakan untuk menyimpan hal-hal yang sifatnya sensitif. Penggunaan komputer untuk menyimpan informasi yang sifatnya classified baru dilakukan di sekitar tahun 1950-an.

Sangat pentingnya nilai sebuah informasi menyebabkan seringkali informasi diinginkan hanya boleh diakses oleh orang-orang tertentu. Jatuhnya informasi ke tangan pihak lain (misalnya pihak lawan bisnis) dapat menimbulkan kerugian bagi pemilik informasi. Sebagai contoh, banyak informasi dalam sebuah perusahaan yang hanya diperbolehkan diketahui oleh orang-orang tertentu di dalam perusahaan tersebut, seperti misalnya informasi tentang produk yang sedang dalam development, algoritma-algoritma dan teknik-teknik yang digunakan untuk menghasilkan produk tersebut. Untuk itu keamanan dari sistem informasi yang digunakan harus terjamin dalam batas yang dapat diterima.

Jaringan komputer, seperti LAN[1] dan Internet, memungkinkan untuk menyediakan informasi secara cepat. Ini salah satu alasan perusahaan atau organisasi mulai berbondong-bondong membuat LAN untuk sistem informasinya dan menghubungkan LAN tersebut ke Internet. Terhubungnya LAN atau komputer ke Internet membuka potensi adanya lubang keamanan (*security hole*) yang tadinya bisa ditutupi dengan mekanisme keamanan secara fisik. Ini sesuai dengan pendapat bahwa kemudahan (kenyamanan) mengakses informasi berbanding terbalik dengan tingkat keamanan sistem informasi itu sendiri. Semakin tinggi tingkat keamanan, semakin sulit (tidak nyaman) untuk mengakses informasi.

Menurut G. J. Simons, keamanan informasi adalah bagaimana kita dapat mencegah penipuan (*cheating*) atau, paling tidak, mendeteksi

1. LAN = Local Area Network

adanya penipuan di sebuah sistem yang berbasis informasi, dimana informasinya sendiri tidak memiliki arti fisik.

Keamanan dan management perusahaan

Seringkali sulit untuk membujuk management perusahaan atau pemilik sistem informasi untuk melakukan investasi di bidang keamanan. Di tahun 1997 majalah Information Week melakukan survey terhadap 1271 *system* atau *network manager* di Amerika Serikat. Hanya 22% yang menganggap keamanan sistem informasi sebagai komponen sangat penting ("*extremely important*"). Mereka lebih mementingkan "*reducing cost*" dan "*improving competitiveness*" meskipun perbaikan sistem informasi setelah dirusak justru dapat menelan biaya yang lebih banyak.

Meskipun sering terlihat sebagai besaran yang tidak dapat langsung diukur dengan uang (*intangible*), keamanan sebuah sistem informasi sebetulnya dapat diukur dengan besaran yang dapat diukur dengan uang (*tangible*). Dengan adanya ukuran yang terlihat, mudah-mudahan pihak management dapat mengerti pentingnya investasi di bidang keamanan. Berikut ini adalah berapa contoh kegiatan yang dapat anda lakukan:

- Hitung kerugian apabila sistem informasi anda tidak bekerja selama 1 jam, selama 1 hari, 1 minggu, dan 1 bulan.
- Hitung kerugian apabila ada kesalahan informasi (data) pada sistem informasi anda. Misalnya web site anda mengumumkan harga sebuah barang yang berbeda dengan harga yang ada di toko anda.
- Hitung kerugian apabila ada data yang hilang, misalnya berapa kerugian yang diderita apabila daftar pelanggan dan invoice hilang dari sistem anda. Berapa biaya yang dibutuhkan untuk rekonstruksi data.

Lawrie Brown dalam [2] menyarankan menggunakan "*Risk Management Model*" untuk menghadapi ancaman (*managing threats*).

Ada tiga komponen yang memberikan kontribusi kepada Risk, yaitu *Asset*, *Vulnerabilities*, dan *Threats*.

TABLE 1. Kontribusi terhadap Risk

Nama komponen	Contoh dan keterangan lebih lanjut
Assets (aset)	• hardware • software • dokumentasi • data • komunikasi • linkungan • manusia
Threats (ancaman)	• pemakai (*users*) • teroris • kecelakaan (*accidents*) • crackers • penjahat kriminal • nasib (*acts of God*) • intel luar negeri (*foreign intelligence*)
Vulnerabilities (kelemahan)	• software bugs • hardware bugs • radiasi (dari layar, transmisi) • tapping, crosstalk • *unauthorized users* • cetakan, *hardcopy* atau print out • keteledoran (*oversight*) • cracker via telepon • storage media

Untuk menanggulangi resiko (*Risk*) tersebut dilakukan apa yang disebut "*countermeasures*" yang dapat berupa:

- usaha untuk mengurangi *Threat*
- usaha untuk mengurangi *Vulnerability*
- usaha untuk mengurangi impak (*impact*)
- mendeteksi kejadian yang tidak bersahabat (*hostile event*)

- kembali (*recover*) dari kejadian

Beberapa Statistik Sistem Keamanan

Ada beberapa statistik yang berhubungan dengan keamanan sistem informasi yang dapat ditampilkan di sini. Data-data yang ditampilkan umumnya bersifat konservatif mengingat banyak perusahaan yang tidak ingin diketahui telah mengalami "security breach" dikarenakan informasi ini dapat menyebabkan "negative publicity". Perusahan-perusahaan tersebut memilih untuk diam dan mencoba menangani sendiri masalah keamanannya tanpa publikasi.

- Tahun 1996, *U.S. Federal Computer Incident Response Capability* (FedCIRC) melaporkan bahwa lebih dari 2500 "insiden" di sistem komputer atau jaringan komputer yang disebabkan oleh gagalnya sistem keamanan atau adanya usaha untuk membobol sistem keamanan [13].
- Juga di tahun 1996, *FBI National Computer Crimes Squad*, Washington D.C., memperkirakan kejahatan komputer yang terdeteksi kurang dari 15%, dan hanya 10% dari angka itu yang dilaporkan [13].
- Sebuah penelitian di tahun 1997 yang dilakukan oleh perusahaan *Deloitte Touch Tohmatsu* menunjukkan bahwa dari 300 perusahaan di Australia, 37% (dua diantara lima) pernah mengalami masalah keamanan sistem komputernya. [16]
- Penelitian di tahun 1996 oleh American Bar Association menunjukkan bahwa dari 1000 perusahaan, 48% telah mengalami "computer fraud" dalam kurun lima tahun terakhir. [16]
- Di Inggris, 1996 *NCC Information Security Breaches Survey* menunjukkan bahwa kejahatan komputer menaik 200% dari tahun 1995 ke 1996. Survey ini juga menunjukkan bahwa kerugian yang diderita rata-rata US $30.000 untuk setiap insiden. Ditunjukkan juga beberapa organisasi yang mengalami kerugian sampai US $1.5 juta.

- FBI melaporkan bahwa kasus persidangan yang berhubungan dengan kejahatan komputer meroket 950% dari tahun 1996 ke tahun 1997, dengan penangkapan dari 4 ke 42, dan terbukti (*convicted*) di pengadilan naik 88% dari 16 ke 30 kasus.
- John Howard dalam penelitiannya di CERT yang belokasi di Carnegie Mellon University mengamati insiden di Internet yang belangsung selama kurun waktu 1989 sampai dengan 1995. Hasil penelitiannya antara lain bahwa setiap domain akan mengalami insiden sekali dalam satu tahun dan sebuah komputer (host) akan mengalami insiden sekali dalam 45 tahun.
- Winter 1999, *Computer Security Institute* dan FBI melakukan survey yang kemudian hasilnya diterbitkan dalam laporannya [5]. Dalam laporan ini terdapat bermacam-macam statistik yang menarik, antara lain bahwa 62% responden merasa bahwa pada 12 bulan terakhir ini ada penggunaan sistem komputer yang tidak semestinya (unauthorized use), 57% merasa bahwa hubungan ke Internet merupakan sumber serangan, dan 86% merasa kemungkinan serangan dari dalam (disgruntled employees) dibandingkan dengan 74% yang merasa serangan dari hackers.

Jebolnys sistem kemanan tentunya membawa dampak. Ada beberapa contoh akibat dari jebolnya sistem keamanan, antara lain:

- 1988. Keamanan sistem mail *sendmail* dieksploitasi oleh Robert Tapan Morris sehingga melumpuhkan sistem Internet. Kegiatan ini dapat diklasifikasikan sebagai "*denial of service attack*". Diperkirakan biaya yang digunakan untuk memperbaiki dan hal-hal lain yang hilang adalah sekitar $100 juta. Di tahun 1990 Morris dihukum (convicted) dan hanya didenda $10.000.
- 10 Maret 1997. Seorang hacker dari Massachusetts berhasil mematikan sistem telekomunikasi di sebuah airport lokal (Worcester, Massachusetts) sehingga mematikan komunikasi di control tower dan menghalau pesawat yang hendak mendarat. Dia juga mengacaukan sistem telepon di Rutland, Massachusetts.
  http://www.news.com/News/Item/Textonly/0,25,20278,00.html?pfv
  http://www.news.com/News/Item/0,4,20226,00.html

Masalah keamanan yang berhubungan dengan Indonesia

Meskipun Internet di Indonesia masih dapat tergolong baru, sudah ada beberapa kasus yang berhubungan dengan keamanan di Indonesia. Di bawah ini akan didaftar beberapa contoh masalah atau topik tersebut.

- **Akhir Januari 1999.** Domain yang digunakan untuk Timor Timur (.TP) diserang sehingga hilang. Domain untuk Timor Timur ini diletakkan pada sebuah server di Irlandia yang bernama *Connect-Ireland*. Pemerintah Indonesia yang disalahkan atau dianggap melakukan kegiatan *hacking* ini. Menurut keterangan yang diberikan oleh administrator Connect-Ireland, 18 serangan dilakukan secara serempak dari seluruh penjuru dunia. Akan tetapi berdasarkan pengamatan, domain Timor Timur tersebut dihack dan kemudian ditambahkan sub domain yang bernama "*need.tp*". Berdasarkan pengamatan situasi, "*need.tp*" merupakan sebuah perkataan yang sedang dipopulerkan oleh "*Beavis and Butthead*" (sebuah acara TV di MTV). Dengan kata lain, crackers yang melakukan serangan tersebut kemungkinan penggemar (atau paling tidak, pernah nonton) acara *Beavis dan Butthead* itu. Jadi, kemungkinan dilakukan oleh seseorang dari Amerika Utara.
- Beberapa web site Indonesia sudah dijebol dan daftarnya (beserta contoh halaman yang sudah dijebol) dapat dilihat di koleksi <http://www.2600.com>

Meningkatnya Kejahatan Komputer

Jumlah kejahatan komputer (*computer crime*), terutama yang berhubungan dengan sistem informasi, akan terus meningkat dikarenakan beberapa hal, antara lain:

- Aplikasi bisnis yang menggunakan (berbasis) teknologi informasi dan jaringan komputer semakin meningkat. Sebagai contoh saat ini mulai bermunculan aplikasi bisnis seperti *on-line banking*, *electronic commerce (e-commerce)*, *Electronic Data Interchange* (EDI), dan masih banyak lainnya. Bahkan aplikasi *e-commerce* akan

menjadi salah satu aplikasi pemacu di Indonesia (melalui "Telematika Indonesia" [29] dan Nusantara 21) dan di seluruh dunia.

- Desentralisasi (dan distributed) server menyebabkan lebih banyak sistem yang harus ditangani. Hal ini membutuhkan lebih banyak operator dan administrator yang handal yang juga kemungkinan harus disebar di seluruh lokasi. Padahal mencari operator dan administrator yang handal adalah sangat sulit.
- Transisi dari single vendor ke multi-vendor sehingga lebih banyak sistem atau perangkat yang harus dimengerti dan masalah interoperability antar vendor yang lebih sulit ditangani. Untuk memahami satu jenis perangkat dari satu vendor saja sudah susah, apalagi harus menangani berjenis-jenis perangkat.
- Meningkatnya kemampuan pemakai di bidang komputer sehingga mulai banyak pemakai yang mencoba-coba bermain atau membongkar sistem yang digunakannya.
- Kesulitan dari penegak hukum untuk mengejar kemajuan dunia komputer dan telekomunikasi yang sangat cepat.
- Semakin kompleksnya sistem yang digunakan, seperti semakin besarnya program (*source code*) yang digunakan sehingga semakin besar probabilitas terjadinya lubang keamanan (yang disebabkan kesalahan pemrograman).
- Semakin banyak perusahaan yang menghubungkan sistem informasinya dengan jaringan komputer yang global seperti Internet. Hal ini membuka akses dari seluruh dunia. Potensi sistem informasi yang dapat dijebol menjadi lebih besar.

Klasifikasi Kejahatan Komputer

Kejahatan komputer dapat digolongkan kepada yang sangat berbahaya sampai ke yang hanya mengesalkan (*annoying*). Menurut David Icove [13] berdasarkan lubang keamanan, keamanan dapat diklasifikasikan menjadi empat, yaitu:

1. **Keamanan yang bersifat fisik** (*physical security*): termasuk akses orang ke gedung, peralatan, dan media yang digunakan. Beberapa bekas penjahat komputer (crackers) mengatakan bahwa mereka sering pergi ke tempat sampah untuk mencari berkas-berkas yang mungkin memiliki informasi tentang keamanan. Misalnya pernah diketemukan coretan password atau manual yang dibuang tanpa dihancurkan. Wiretapping atau hal-hal yang berhubungan dengan akses ke kabel atau komputer yang digunakan juga dapat dimasukkan ke dalam kelas ini.
   Denial of service, yaitu akibat yang ditimbulkan sehingga servis tidak dapat diterima oleh pemakai juga dapat dimasukkan ke dalam kelas ini. *Denial of service* dapat dilakukan misalnya dengan mematikan peralatan atau membanjiri saluran komunikasi dengan pesan-pesan (yang dapat berisi apa saja karena yang diutamakan adalah banyaknya jumlah pesan). Beberapa waktu yang lalu ada lubang keamanan dari implementasi protokol TCP/IP yang dikenal dengan istilah *Syn Flood Attack*, dimana sistem (*host*) yang dituju dibanjiri oleh permintaan sehingga dia menjadi terlalu sibuk dan bahkan dapat berakibat macetnya sistem (*hang*).
2. **Keamanan yang berhubungan dengan orang (personel)**: termasuk identifikasi, dan profil resiko dari orang yang mempunyai akses (pekerja). Seringkali kelemahan keamanan sistem informasi bergantung kepada manusia (pemakai dan pengelola). Ada sebuah teknik yang dikenal dengan istilah "*social engineering*" yang sering digunakan oleh kriminal untuk berpura-pura sebagai orang yang berhak mengakses informasi. Misalnya kriminal ini berpura-pura sebagai pemakai yang lupa passwordnya dan minta agar diganti menjadi kata lain.
3. **Keamanan dari data dan media serta teknik komunikasi** (*communications*). Yang termasuk di dalam kelas ini adalah kelemahan dalam software yang digunakan untuk mengelola data. Seorang kriminal dapat memasang *virus* atau *trojan horse* sehingga dapat mengumpulkan informasi (seperti password) yang semestinya tidak berhak diakses.
4. **Keamanan dalam operasi**: termasuk prosedur yang digunakan untuk mengatur dan mengelola sistem keamanan, dan juga termasuk prosedur setelah serangan (*post attack recovery*).

Aspek / servis dari security

A computer is secure if you can depend on it and its software to behave as you expect. (Garfinkel and Spafford)

Garfinkel [11] mengemukakan bahwa keamanan komputer *(computer security*) melingkupi empat aspek, yaitu privacy, integrity, authentication, dan availability. Selain keempat hal di atas, masih ada dua aspek lain yang juga sering dibahas dalam kaitannya dengan *electronic commerce*, yaitu access control dan non-repudiation.

Privacy / Confidentiality

Inti utama aspek *privacy* atau *confidentiality* adalah usaha untuk menjaga informasi dari orang yang tidak berhak mengakses. Privacy lebih kearah data-data yang sifatnya privat sedangkan confidentiality biasanya berhubungan dengan data yang diberikan ke pihak lain untuk keperluan tertentu (misalnya sebagai bagian dari pendaftaran sebuah servis) dan hanya diperbolehkan untuk keperluan tertentu tersebut. Contoh hal yang berhubungan dengan *privacy* adalah e-mail seorang pemakai (*user*) tidak boleh dibaca oleh administrator. Contoh *confidential information* adalah data-data yang sifatnya pribadi (seperti nama, tempat tanggal lahir, social security number, agama, status perkawinan, penyakit yang pernah diderita, nomor kartu kredit, dan sebagainya) merupakan data-data yang ingin diproteksi penggunaan dan penyebarannya. Contoh lain dari confidentiality adalah daftar pelanggan dari sebuah *Internet Service Provider* (ISP).

Serangan terhadap aspek privacy misalnya adalah usaha untuk melakukan penyadapan (dengan program *sniffer*). Usaha-usaha yang dapat dilakukan untuk meningkatkan privacy dan confidentiality adalah dengan menggunakan teknologi kriptografi.

Integrity

Aspek ini menekankan bahwa informasi tidak boleh diubah tanpa seijin pemilik informasi. Adanya virus, *trojan horse*, atau pemakai lain yang mengubah informasi tanpa ijin merupakan contoh masalah yang harus dihadapi. Sebuah e-mail dapat saja "ditangkap" (*intercept*) di tengah jalan, diubah isinya, kemudian diteruskan ke alamat yang dituju. Dengan kata lain, integritas dari informasi sudah tidak terjaga. Penggunaan enkripsi dan digital signature, misalnya, dapat mengatasi masalah ini.

Salah satu contoh kasus trojan horse adalah distribusi paket program *TCP Wrapper* (yaitu program populer yang dapat digunakan untuk mengatur dan membatasi akses TCP/IP) yang dimodifikasi oleh orang yang tidak bertanggung jawab. Jika anda memasang program yang berisi trojan horse tersebut, maka ketika anda merakit (compile) program tersebut, dia akan mengirimkan eMail kepada orang tertentu yang kemudian memperbolehkan dia masuk ke sistem anda. Informasi ini berasal dari CERT Advisory, "*CA-99-01 Trojan-TCP-Wrappers*" yang didistribusikan 21 Januari 1999. Contoh serangan lain adalah yang disebut "man in the middle attack" dimana seseorang menempatkan diri di tengah pembicaraan dan menyamar sebagai orang lain.

Authentication

Aspek ini berhubungan dengan metoda untuk menyatakan bahwa informasi betul-betul asli, atau orang yang mengakses atau memberikan informasi adalah betul-betul orang yang dimaksud. Masalah pertama, membuktikan keaslian dokumen, dapat dilakukan dengan teknologi *watermarking* dan digital signature. Watermarking juga dapat digunakan untuk menjaga "*intelectual property*", yaitu dengan menandai dokumen atau hasil karya dengan "tanda tangan" pembuat . Masalah kedua biasanya berhubungan dengan access control, yaitu berkaitan dengan pembatasan orang yang dapat mengakses informasi. Dalam hal ini pengguna harus menunjukkan bukti bahwa memang dia adalah pengguna yang sah,

misalnya dengan menggunakan password, biometric (ciri-ciri khas orang), dan sejenisnya. Penggunaan teknologi *smart card*, saat ini kelihatannya dapat meningkatkan keamanan aspek ini.

Availability

Aspek availability atau ketersediaan berhubungan dengan ketersediaan informasi ketika dibutuhkan. Sistem informasi yang diserang atau dijebol dapat menghambat atau meniadakan akses ke informasi. Contoh hambatan adalah serangan yang sering disebut dengan "*denial of service attack*" (DoS attack), dimana server dikirimi permintaan (biasanya palsu) yang bertubi-tubi atau permintaan yang diluar perkiraan sehingga tidak dapat melayani permintaan lain atau bahkan sampai *down*, *hang*, *crash*. Contoh lain adalah adanya *mailbomb*, dimana seorang pemakai dikirimi e-mail bertubi-tubi (katakan ribuan e-mail) dengan ukuran yang besar sehingga sang pemakai tidak dapat membuka e-mailnya atau kesulitan mengakses e-mailnya (apalagi jika akses dilakukan melalui saluran telepon). Bayangkan apabila anda dikirimi 5000 email dan anda harus mengambil (download) email tersebut melalui telepon dari rumah.

Serangan terhadap availability dalam bentuk DoS attack merupakan yang terpopuler pada saat naskah ini ditulis. Pada bagian lain akan dibahas tentang serangan DoS ini secara lebih rinci. (Lihat "Denial of Service Attack" pada halaman 69.)

Access Control

Aspek ini berhubungan dengan cara pengaturan akses kepada informasi. Hal ini biasanya berhubungan dengan masalah authentication dan juga privacy. Access control seringkali dilakukan dengan menggunakan kombinasi userid / password atau dengan menggunakan mekanisme lain.

Non-repudiation

Aspek ini menjaga agar seseorang tidak dapat menyangkal telah melakukan sebuah transaksi. Sebagai contoh, seseorang yang mengirimkan email untuk memesan barang tidak dapat menyangkal bahwa dia telah mengirimkan email tersebut. Aspek ini sangat penting dalam hal electronic commerce. Penggunaan digital signature dan teknologi kriptografi secara umum dapat menjaga aspek ini. Akan tetapi hal ini masih harus didukung oleh hukum sehingga status dari digital signature itu jelas legal. Hal ini akan dibahas lebih rinci pada bagian tersendiri.

Serangan Terhadap Keamanan Sistem Informasi

Security attack, atau serangan terhadap keamanan sistem informasi, dapat dilihat dari sudut peranan komputer atau jaringan komputer yang fungsinya adalah sebagai penyedia informasi. Menurut W. Stallings [27] ada beberapa kemungkinan serangan (*attack*):

- *Interruption*: Perangkat sistem menjadi rusak atau tidak tersedia. Serangan ditujukan kepada ketersediaan (*availability*) dari sistem. Contoh serangan adalah “denial of service attack”.
- *Interception*: Pihak yang tidak berwenang berhasil mengakses aset atau informasi. Contoh dari serangan ini adalah penyadapan (*wiretapping*).
- *Modification*: Pihak yang tidak berwenang tidak saja berhasil mengakses, akan tetapi dapat juga mengubah (tamper) aset. Contoh dari serangan ini antara lain adalah mengubah isi dari web site dengan pesan-pesan yang merugikan pemilik web site.
- *Fabrication*: Pihak yang tidak berwenang menyisipkan objek palsu ke dalam sistem. Contoh dari serangan jenis ini adalah memasukkan pesan-pesan palsu seperti e-mail palsu ke dalam jaringan komputer.

Electronic commerce: mengapa sistem informasi berbasis Internet

Sistem informasi saat ini banyak yang mulai menggunakan basis Internet. Ini disebabkan Internet merupakan sebuah platform yang terbuka (*open platform*) sehingga menghilangkan ketergantungan perusahaan pada sebuah vendor tertentu seperti jika menggunakan sistem yang tertutup (*proprietary systems*). Open platform juga mempermudah interoperability antar vendor.

Selain alasan di atas, saat ini Internet merupakan media yang paling ekonomis untuk digunakan sebagai basis sistem informasi. Hubungan antar komputer di Internet dilakukan dengan menghubungkan diri ke link terdekat, sehingga hubungan fisik biasanya bersifat lokal. Perangkat lunak (*tools*) untuk menyediakan sistem informasi berbasis Internet (dalam bentuk server web, ftp, gopher), membuat informasi (HTML editor), dan untuk mengakses informasi (web browser) banyak tersedia. Perangkat lunak ini banyak yang tersedia secara murah dan bahkan gratis.

Alasan-alasan tersebut di atas menyebabkan Internet menjadi media elektronik yang paling populer untuk menjalankan bisnis, yang kemudian dikenal dengan istilah electronic commerce (e-commerce). Dengan diperbolehkannya bisnis menggunakan Internet, maka penggunaan Internet menjadi meledak. Statistik yang berhubungan dengan kemajuan Internet dan e-commerce sangat menakjubkan.

Statistik Internet

Jumlah komputer, server, atau lebih sering disebut *host* yang terdapat di Internet menaik dengan angka yang fantastis. Sejak tahun 1985 sampai dengan tahun 1997 tingkat perkembangannya (*growth rate*) jumlah host setiap tahunnya adalah 2,176. Jadi setiap tahun jumlah host meningkat lebih dari dua kali. Pada saat naskah ini ditulis (akhir tahun 1999), growth rate sudah turun menjadi 1,5.

Data-data statistik tentang pertumbuhan jumlah host di Internet dapat diperoleh di "Matrix Maps Quarterly" yang diterbitkan oleh MIDS[1]. Beberapa fakta menarik tentang Internet:

- Jumlah host di Internet Desember 1969: 4
- Jumlah host di Internet Agustus 1981: 213
- Jumlah host di Internet Oktober 1989: 159.000
- Jumlah host di Internet Januari 1992: 727.000

Statistik Electronic Commerce

Hampir mirip dengan statistik jumlah host di Internet, statistik penggunaan Internet untuk keperluan e-commerce juga meningkat dengan nilai yang menakjubkan. Berikut ini adalah beberapa data yang diperoleh dari International Data Corporation (IDC):

- Perkiraan pembelian konsumer melalui Web di tahun 1999: US$ 31 billion (31 milyar dolar Amerika). Diperkirakan pada tahun 2003 angka ini menjadi US$177,7 billion.
- Perkiraan pembelian bisnis melalui web di tahun 1999: US$80,4 billion (80,4 milyar dolar Amerika). Diperkirakan pada tahun 2003 angka ini menjadi US$1.1 trillion.
- Jika diperhatikan angka-angka di atas, maka e-commerce yang sifatnya bisnis (*business to business*) memiliki nilai yang lebih besar dibandingkan yang bersifat *business to consumer*.

Di Indonesia, e-commerce merupakan sebuah tantangan yang perlu mendapat perhatian lebih serius. Ada beberapa hambatan dan juga peluang di dalam bidang ini. Pembahasan tentang e-commerce di Indonesia dapat dilihat di [17, 23].

1. http://www.mids.org

Keamanan Sistem Internet

Untuk melihat keamanan sistem Internet perlu diketahui cara kerja sistem Internet. Antara lain, yang perlu diperhatikan adalah hubungan antara komputer di Internet, dan protokol yang digunakan. Internet merupakan jalan raya yang dapat digunakan oleh semua orang (*public*). Untuk mencapai server tujuan, paket informasi harus melalui beberapa sistem (router, gateway, hosts, atau perangkat-perangkat komunikasi lainnya) yang kemungkinan besar berada di luar kontrol dari kita. Setiap titik yang dilalui memiliki potensi untuk dibobol, disadap, dipalsukan [22]. Kelemahan sebuat sistem terletak kepada komponen yang paling lemah.

Asal usul Internet kurang memperhatikan masalah keamanan. Ini mungkin dikarenakan unsur kental dari perguruan tinggi dan lembaga penelitian yang membangun Internet. Sebagai contoh, IP versi 4 yang digunakan di Internet banyak memiliki kelemahan. Hal ini dicoba diperbaiki dengan IP Secure dan IP versi 6.

Hackers, Crackers, dan Etika

Hackers are like kids putting a 10 pence piece on a railway line to see if the train can bend it, not realising that they risk de-railing the whole train (Mike Jones: London interview).

Untuk mempelajari masalah keamanan, ada baiknya juga mempelajari aspek dari pelaku yang terlibat dalam masalah keamanan ini, yaitu para hackers and crackers. Buku ini tidak bermaksud untuk membahas secara terperinci masalah non-teknis (misalnya sosial) dari hackers akan tetapi sekedar memberikan ulasan singkat.

Hackers vs crackers

hacker /n./

> [originally, someone who makes furniture with an axe] 1. A person who enjoys exploring the details of programmable systems and how to stretch their capabilities, as opposed to most users, who prefer to learn only the minimum necessary. 2. One who programs enthusiastically (even obsessively) or who enjoys programming rather than just theorizing about programming. 3. A person capable of appreciating hack value. 4. A person who is good at programming quickly. 5. An expert at a particular program, or one who frequently does work using it or on it; as in `a Unix hacker'. (Definitions 1 through 5 are correlated, and people who fit them congregate.) 6. An expert or enthusiast of any kind. One might be an astronomy hacker, for example. 7. One who enjoys the intellectual challenge of creatively overcoming or circumventing limitations. 8. [deprecated] A malicious meddler who tries to discover sensitive information by poking around. Hence `password hacker', `network hacker'. The correct term for this sense is cracker.

Istilah hackers sendiri masih belum baku karena bagi sebagian orang hackers mempunyai konotasi positif, sedangkan bagi sebagian lain memiliki konotasi negatif. Bagi kelompok yang pertama (*old school*), untuk pelaku yang jahat biasanya disebut *crackers*. Batas antara hacker dan cracker sangat tipis. Batasan ini ditentukan oleh etika. moral, dan integritas dari pelaku sendiri. Untuk selanjutnya dalam buku ini kami akan menggunakan kata hacker sebagai generalisir dari hacker dan cracker, kecuali bila diindikasikan secara eksplisit.

Paul Taylor dalam disertasi PhDnya [28] mengungkapkan adanya tiga kelompok, yaitu *Computer Underground* (CU), *Computer Security Industry* (CSI), dan kelompok akademis. Perbedaan antar kelompok ini kadang-kadang tidak tegas.

Untuk sistem yang berdomisili di Indonesia secara fisik (*physical*) maupun lojik (*logical*) ancaman keamanan dapat datang dari berbagai pihak. Berdasarkan sumbernya, acaman dapat dikategorikan yang berasal dari luar negeri dan yang berasal dari dalam negeri. Acaman yang berasal dari luar negeri contohnya adalah hackers Portugal yang mengobrak-abrik beberapa web site milik pemerintah Indonesia.

Berdasarkan motif dari para perusak, ada yang berbasis politik, eknomi, dan ada juga yang hanya ingin mencari ketenaran. Masalah politik nampaknya sering menjadi alasan untuk menyerang sebuah

sistem. Beberapa contoh dari serangan yang menggunakan alasan politik antara lain:

- Serangan dari hackers Portugal yang mengubah isi beberapa web site milik pemerintah Indonesia dikarenakan hackers tersebut tidak setuju dengan apa yang dilakukan oleh pemerintah Indonesia di Timor Timur. Selain mengubah isi web site, mereka juga mencoba merusak sistem yang ada dengan menghapus seluruh disk (jika bisa).
- Serangan dari hackers Cina dan Taiwan terhadap beberapa web site Indonesia atas kerusuhan di Jakarta (Mei 1998) yang menyebabkan etnis Cina di Indonesia mendapat perlakukan yang tidak adil. Hackers ini mengubah beberapa web site Indonesia untuk menyatakan ketidak-sukaan mereka atas apa yang telah terjadi.
- Beberapa hackers di Amerika menyatakan akan merusak sistem milik pemerintah Iraq ketika terjeadi ketegangan politik antara Amerika dan Irak.

Interpretasi Etika Komputasi

Salah satu hal yang membedakan antara crackers dan hackers, atau antara Computer Underground dan Computer Security Industry adalah masalah etika. Keduanya memiliki basis etika yang berbeda atau mungkin memiliki interpretasi yang berbeda terhadap suatu topik yang berhubungan dengan masalah computing. Kembali, Paul Taylor melihat hal ini yang menjadi basis pembeda keduanya. Selain masalah kelompok, kelihatannya umur juga membedakan pandangan (interpretasi) terhadap suatu topik. Salah satu contoh, Computer Security Industry beranggapan bahwa Computer Underground masih belum memahami bahwa "*computing*" tidak sekedar permainan dan mereka (maksudnya CU) harus melepaskan diri dari "*playpen*[1]".

1. playpen = boks tempat bayi bermain

Perbedaan pendapat ini dapat muncul di berbagai topik. Sebagai contoh, bagaimana pendapat anda tentang memperkerjakan seorang hacker sebagai kepala keamanan sistem informasi anda? Ada yang berpendapat bahwa hal ini sama dengan memperkerjakan penjarah (gali, preman) sebagai kepala keamanan setempat. Jika analogi ini disepakati, maka akibat negatif yang ditimbulkan dapat dimengerti. Akan tetapi para computer underground berpendapat bahwa analogi tersebut kurang tepat. Para computer underground berpendapat bahwa hacking lebih mengarah ke kualitas intelektual dan jiwa pionir. Kalau dianalogikan, mungkin lebih ke arah permainan catur dan masa "*wild west*" (di Amerika jaman dahulu). Pembahasan yang lebih detail tentang hal ini dapat dibaca dalam disertasi dari Paul Taylor [28].

Perbedaan pendapat juga terjadi dalam masalah "*probing*", yaitu mencari tahu kelemahan sebuah sistem. Computer security industry beranggapan bahwa probing merupakan kegiatan yang tidak etis. Sementara para computer underground menganggap bahwa mereka membantu dengan menunjukkan adanya kelemahan dalam sebuah sistem (meskipun sistem tersebut bukan dalam pengelolaannya). Kalau dianalogikan ke dalam kehidupan sehari-hari (jika anda setuju dengan analoginya), bagaimana pendapat anda terhadap seseorang (yang tidak diminta) yang mencoba-coba membuka-buka pintu atau jendela rumah anda dengan alasan untuk menguji keamanan rumah anda.

Hackers dan crackers Indonesia

Apakah ada hackers dan crackers Indonesia? Tentunya ada. Kedua "school of thought" (madzhab) hackers ada di Indonesia. Kelompok yang menganut "old school" dimana hacking tidak dikaitkan dengan kejahatan elektronik umumnya bergabung di berbagai mailing list dan kelompok baik secara terbuka maupun tertutup. Ada beberapa mailing list dimana para hackers bergabung, antara lain:

- Mailing list pau-mikro. Mailing list ini mungkin termasuk yang tertua di Indonesia, dimulai sejak akhir tahun 1980-an oleh yang sedang bersekolah di luar negeri (dimotori oleh staf PAU Mikroelektronika ITB dimana penulis merupakan salah satu motornya, yang kemudian malah menjadi minoritas di milis tersebut). Milis ini tadinya berkedudukan di jurusan elektro University of Manitoba, Canada (sehingga memiliki alamat pau-mikro@ee.umanitoba.ca) dan kemudian pindah menjadi pau-mikro@nusantara.net.
- Hackerlink
- Kecoa Elektronik yang memiliki homepage di <http://k-elektronik.org>

Selain tempat berkumpul hacker, ada juga tempat profesional untuk menjalankan security seperti di

- IDCERT - Indonesia Computer Emergency Response Team
  http://www.cert.or.id

BAB 2

Dasar-Dasar Keamanan Sistem Informasi

Sebelum melangkah lebih jauh kepada hal yang praktis dalam pengamanan sistem informasi, ada baiknya kita pelajari dasar-dasar (principles) dan teori-teori yang digunakan untuk pengamanan sistem informasi. Kriptografi dan enkripsi (baik dengan menggunakan private-key maupun dengan menggunakan public-key) akan dibahas di dalam bab ini.

Terminologi

Kriptografi (*cryptography*) merupakan ilmu dan seni untuk menjaga pesan agar aman. (*Cryptography is the art and science of keeping messages secure.* [27]) "*Crypto*" berarti "*secret*" (rahasia) dan "*graphy*" berarti "*writing*" (tulisan) [2]. Para pelaku atau praktisi kriptografi disebut ***cryptographers***. Sebuah algoritma kriptografik (*cryptographic algorithm*), disebut **cipher**, merupakan persamaan matematik yang digunakan untuk proses enkripsi dan dekripsi. Biasanya kedua persamaan matematik (untuk enkripsi dan dekripsi) tersebut memiliki hubungan matematis yang cukup erat.

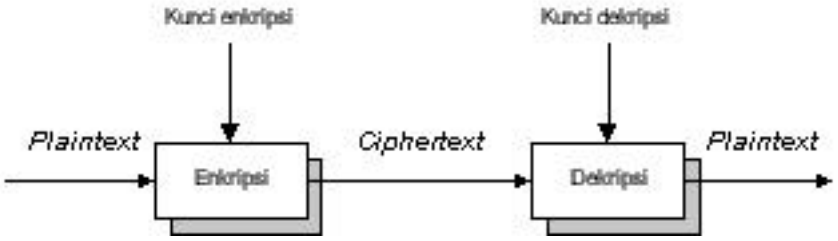
Proses yang dilakukan untuk mengamankan sebuah pesan (yang disebut *plaintext*) menjadi pesan yang tersembunyi (disebut *ciphertext*) adalah **enkripsi** (*encryption*). *Ciphertext* adalah pesan yang sudah tidak dapat dibaca dengan mudah. Menurut ISO 7498-2, terminologi yang lebih tepat digunakan adalah "*encipher*".

Proses sebaliknya, untuk mengubah *ciphertext* menjadi *plaintext*, disebut **dekripsi** (*decryption*). Menurut ISO 7498-2, terminologi yang lebih tepat untuk proses ini adalah "*decipher*".

Cryptanalysis adalah seni dan ilmu untuk memecahkan *ciphertext* tanpa bantuan kunci. *Cryptanalyst* adalah pelaku atau praktisi yang menjalankan *cryptanalysis.*

Enkripsi

Enkripsi digunakan untuk menyandikan data-data atau informasi sehingga tidak dapat dibaca oleh orang yang tidak berhak. Dengan enkripsi data anda disandikan (*encrypted*) dengan menggunakan sebuah kunci (*key*). Untuk membuka (*decrypt*) data tersebut digunakan juga sebuah kunci yang dapat sama dengan kunci untuk mengenkripsi (untuk kasus *private key cryptography*) atau dengan kunci yang berbeda (untuk kasus *public key cryptography*). Gambar 2.1 pada halaman 22 menunjukkan contoh proses enkripsi dan dekripsi dengan dua kunci yang berbeda.



GAMBAR 2.1. Diagram proses enkripsi dan dekripsi

Secara matematis, proses atau fungsi enkripsi (*E*) dapat dituliskan sebagai:

$$E(M) = C \quad \textbf{(1)}$$

dimana: *M* adalah *plaintext* (*message*) dan *C* adalah *ciphertext*.

Proses atau fungsi dekripsi (*D*) dapat dituliskan sebagai:

$$D(C) = M \quad \textbf{(2)}$$

Elemen dari Enkripsi

Ada beberapa elemen dari enkripsi yang akan dijabarkan dalam beberapa paragraf di bawah ini.

Algoritma dari Enkripsi dan Dekripsi. Algoritma dari enkripsi adalah fungsi-fungsi yang digunakan untuk melakukan fungsi enkripsi dan dekripsi. Algoritma yang digunakan menentukan kekuatan dari enkripsi, dan ini biasanya dibuktikan dengan basis matematika.

Kunci yang digunakan dan panjangnya kunci. Kekuatan dari penyandian bergantung kepada kunci yang digunakan. Beberapa algoritma enkripsi memiliki kelemahan pada kunci yang digunakan. Untuk itu, kunci yang lemah tersebut tidak boleh digunakan. Selain itu, panjangnya kunci, yang biasanya dalam ukuran *bit*, juga menentukan kekuatan dari enkripsi. Kunci yang lebih panjang biasanya lebih aman dari kunci yang pendek. Jadi enkripsi dengan menggunakan kunci 128-bit lebih sukar dipecahkan dengan algoritma enkripsi yang sama tetapi dengan kunci 56-bit. Semakin panjang sebuah kunci, semakin besar keyspace yang harus dijalani untuk mencari kunci dengan cara *brute force attack* atau coba-coba karena keyspace yang harus dilihat merupakan pangkat dari bilangan 2. Jadi kunci 128-bit memiliki keyspace 2^{128}, sedangkan kunci 56-bit memiliki keyspace 2^{56}. Artinya semakin lama kunci baru bisa ketahuan.

Plaintext. Plaintext adalah pesan atau informasi yang dikirimkan.

Ciphertext. Ciphertext adalah informasi yang sudah dienkripsi.

Kembali ke masalah algoritma, keamanan sebuah algoritma yang digunakan dalam enkripsi atau dekripsi bergantung kepada beberapa aspek. Salah satu aspek yang cukup penting adalah sifat algoritma yang digunakan. Apabila kekuatan dari sebuah algoritma sangat tergantung kepada pengetahuan (tahu atau tidaknya) orang terhadap algoritma yang digunakan, maka algoritma tersebut disebut "restricted algorithm". Apabila algoritma tersebut bocor atau ketahuan oleh orang banyak, maka pesan-pesan dapat terbaca. Tentunya hal ini masih bergantung kepada adanya kriptografer yang baik. Jika tidak ada yang tahu, maka sistem tersebut dapat dianggap aman (meskipun semu).

Meskipun kurang aman, metoda pengamanan dengan *restricted algorithm* ini cukup banyak digunakan karena mudah implementasinya dan tidak perlu diuji secara mendalam. Contoh penggunaan metoda ini adalah enkripsi yang menggantikan huruf yang digunakan untuk mengirim pesan dengan huruf lain. Ini disebut dengan "*substitution cipher*".

Substitution Cipher dengan Caesar Cipher

Salah satu contoh dari "*substitution cipher*" adalah Caesar Cipher yang digunakan oleh Julius Caesar. Pada prinsipnya, setiap huruf digantikan dengan huruf yang berada tiga (3) posisi dalam urutan alfabet. Sebagai contoh huruf "a" digantikan dengan huruf "D" dan seterusnya. Transformasi yang digunakan adalah:

```
plain : a b c d e f g h i j k l m n o p q r s t u v w x y z
cipher: D E F G H I J K L M N O P Q R S T U V W X Y Z A B C
```

Latihan 1. Buat ciphertext dari kalimat di bawah ini.

PESAN SANGAT RAHASIA

Latihan 2. Cari plaintext dari kalimat ini

PHHW PH DIWHU WKH WRJD SDUWB

ROT13

Substitution cipher yang masih umum digunakan di sistem UNIX adalah ROT13. Pada sistem ini sebuah huruf digantikan dengan huruf yang letaknya 13 posisi darinya. Sebagai contoh, huruf "A" digantikan dengan huruf "N", huruf "B" digantikan dengan huruf "O", dan seterusnya. Secara matematis, hal ini dapat dituliskan sebagai:

$$C = ROT13(M) \quad \textbf{(3)}$$

Untuk mengembalikan kembali ke bentuk semulanya dilakukan proses enkripsi ROT13 dua kali [26].

$$M = ROT13(ROT13(M)) \quad \textbf{(4)}$$

ROT13 memang tidak didisain untuk keamanan tingkat tinggi. ROT13, misalnya digunakan untuk menyelubungi isi dari artikel (*posting*) di *Usenet news* yang berbau ofensif. Sehingga hanya orang yang betul-betul ingin membaca dapat melihat isinya. Contoh penggunaan lain adalah untuk menutupi jawaban dari sebuah teka teki (*puzzle*).

Program dalam bahasa *Perl* untuk melakukan ROT13 dapat dilihat dalam listing di bawah ini.

```
#!  /usr/bin/perl
# rot13: rotate 13
# usageL rot13 < filename.txt
# bugs: only works with lower case
#
# Copyright 1998, Budi Rahardjo
# <rahard@paume.itb.ac.id>, <budi@vlsi.itb.ac.id>
# Electrical Engineering
# Institut Teknologi Bandung (ITB), Indonesia
#

while (<>) {
   # read a line into $_
```

```
   for ($i=0 ; $i < length($_) ; $i++) {
      $ch = substr($_,$i,1);
      # only process if it's within a-z
      # otherwise skip
      if ( (ord($ch)>=97) and (ord($ch)<=122) ) {
         $newch = &rot13($ch); # rotate it
         printf("%c", $newch);
      } else {
      # just print character that was not processed
         print $ch;
      }
   }  # end for loop
} # done...

sub rot13 {
   local($ch) = @_;
      $asch = ord($ch) - 97;  # get the ascii value and normalize it
      $rotasch = $asch + 13;  # rotate 13 it
      # send it back to ascii
      $rotasch = $rotasch % 26;
      $rotasch = $rotasch + 97;
      return($rotasch);
}
```

Latihan 3. Gunakan program di atas atau buat program sendiri untuk meng-ROT13-kan kalimat di bawah ini:
"kalau mau aman, pakai enkripsi bung"
Catatan: lupakan spasi dan tanda koma.
Setelah itu, jalankan ROT13 kembali untuk mengembalikan teks menjadi kalimat semula.

Caesar cipher dan ROT13 disebut juga "*monoalphabetic ciphers*" karena setiap huruf digantikan dengan sebuah huruf. Mono alphabetic cipher ini agak mudah dipecahkan dengan menganalisa ciphertext apabila beberapa informasi lain (seperti bahasa yang digunakan) dapat diketahui. Salah satu cara penyerangan (*attack*) yang dapat dilakukan adalah dengan menganalisa statistik dari huruf yang muncul. Stallings dalam bukunya [27] menunjukkan statistik kemunculan huruf untuk tulisan dalam bahasa Inggris. Cara yang sama dapat dilakukan untuk mencari distribusi penggunaan huruf dalam teks berbahasa Indonesia.

```
#!  /usr/bin/perl
# statistik munculnya jumlah huruf
# statchar.pl < filename.txt
# bugs: only works with lower case
#
# Copyright 1998, Budi Rahardjo
# <rahard@paume.itb.ac.id>, <budi@vlsi.itb.ac.id>
# Electrical Engineering
```

```
# Institut Teknologi Bandung (ITB), Indonesia
#

while (<>) {
   # read a line into $_
   for ($i=0 ; $i < length($_) ; $i++) {
      $ch = substr($_,$i,1);
      # only process if it's within a-z
      # otherwise skip
      if ( (ord($ch)>=97) and (ord($ch)<=122) ) {
         $ordch= ord($ch);
         $cumulative{$ordch}++;
         $total++;
      }
   }  # end for loop
} # done...

for ($i=97 ; $i <=122 ; $i++) {
   $muncul = $cumulative{$i};
   $persenmuncul = $muncul / $total * 100;
   printf("%c = %d (%.2g\%)\n", $i, $muncul, $persenmuncul);
}
```

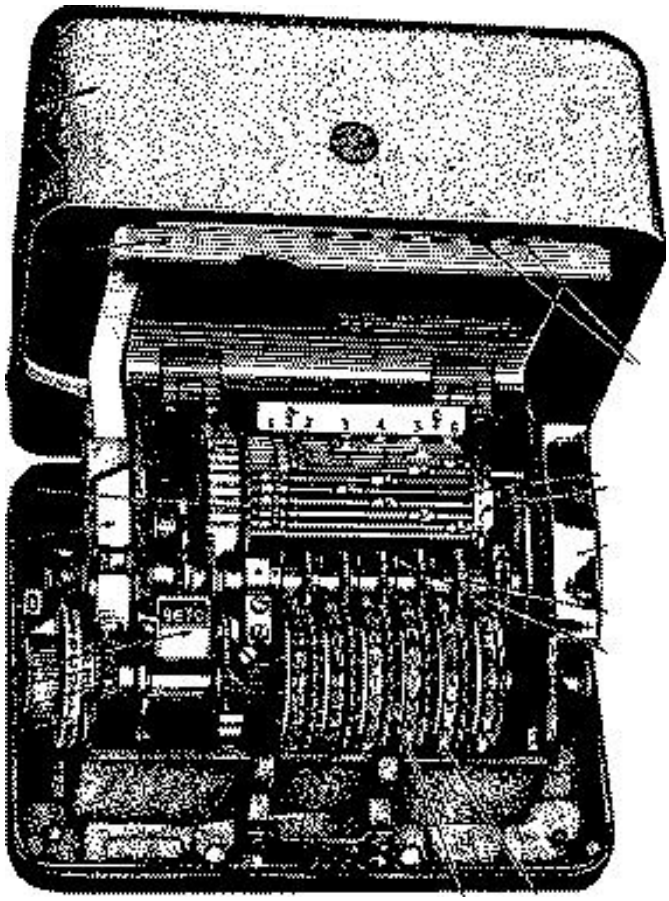
Latihan 4. Cari frekuensi munculnya huruf "a" sampai dengan "z" dalam teks yang menggunakan bahasa Indonesia. Peragakan grafik distribusinya. Buat program sendiri atau gunakan perl script di atas untuk mencari distribusinya.

Multiple-letter encryption

Untuk meningkatkan keamanan, enkripsi dapat dilakukan dengan mengelompokkan beberapa huruf menjadi sebuah kesatuan (unit) yang kemudian dienkripsi. Ini disebut *multiple-letter encryption*. Salah satu contoh multiple-letter encryption adalah "*Playfair*".

Enigma Rotor Machine

Enigma rotor machine merupakan sebuah alat enkripsi yang digunakan dalam perang dunia ke dua. Dia terdiri atas beberapa rotor dan kabel yang silang menyilang menyebabkan substitusi alfabet yang selalu berubah.



Enigma Rotor Machine

Penggunaan Kunci

Salah satu cara untuk menambah tingkat keamanan sebuah algoritma enkripsi dan dekripsi adalah dengan menggunakan sebuah kunci (*key*) yang biasanya disebut *K*. Kunci *K* ini dapat memiliki rentang (*range*) yang cukup lebar. Rentang dari kemungkinan angka (harga) dari kunci *K* ini disebut *keyspace*. Kunci

K ini digunakan dalam proses enkripsi dan dekripsi sehingga persamaan matematisnya menjadi:

$$E_K(M) = C \quad \textbf{(5)}$$

$$D_K(M) = M \quad \textbf{(6)}$$

Keamanan sistem yang digunakan kemudian tidak bergantung kepada pengetahuan algoritma yang digunakan, melainkan bergantung kepada kunci yang digunakan. Artinya, algoritma dapat diketahui oleh umum atau dipublikasikan. Usaha untuk memecahkan keamanan sistem menjadi usaha untuk memecahkan atau mencari kunci yang digunakan.

Usaha mencari kunci sangat bergantung kepada keyspace dari kunci *K*. Apabila keyspace ini cukup kecil, maka cara *brute force* atau mencoba semua kunci dapat dilakukan. Akan tetapi apabila keyspace dari kunci yang digunakan cukup besar, maka usaha untuk mencoba semua kombinasi kunci menjadi tidak realistis. Keyspace dari *DES*, misalnya, memiliki 56-bit. Untuk mencoba semua kombinasi yang ada diperlukan 2^{56} kombinasi. (Cerita tentang kelemahan DES akan diutarakan di bagian lain.)

Latihan 5. Jika sebuah komputer dapat mencoba 1000 kombinasi dalam 1 detik, berapa waktu yang dibutuhkan untuk mencoba semua kombinasi DES yang menggunakan 56 bit?

Aplikasi dari Enkripsi

Contoh penggunaan enkripsi adalah program Pretty Good Privacy (PGP) [11], dan secure shell (SSH). Program PGP digunakan untuk mengenkripsi dan menambahkan *digital siganture* dalam e-mail yang dikirim. Program SSH digunakan untuk mengenkripsi sesion *telnet* ke sebuah host. Hal ini akan dibahas lebih lanjut pada bagian lain.

Public-key cryptography lawan symmetric cryptography

Perbedaan prinsip dan penggunaan *public-key cryptography* dan *symmetric cryptography* membutuhkan diskusi tersendiri. Pada *symmetric cryptography*, satu kunci yang sama digunakan untuk melakukan enkripsi dan dekripsi. Pada sistem *public-key cryptography*, enkripsi dan dekripsi menggunakan kunci yang berbeda.

Sejak dikembangkannya *public-key cryptography*, selalu timbul pertanyaan mana yang lebih baik. Para pakar kriptografi mengatakan bahwa keduanya tidak dapat dibandingkan karena mereka memecahkan masalah dalam domain yang berbeda. *Symmetric cryptography* merupakan hal yang terbaik untuk mengenkripsi data. Kecepatannya dan keamanan akan *choosen-ciphertext attack* merupakan kelebihannya. Sementara itu *public-key cryptography* dapat melakukan hal-hal lain lebih baik daripada *symmetric cryptography*, misalnya dalam hal key management. (Diskusi lebih jauh dapat dilihat di referensi [26].)

Data Encryption Standard (DES)

DES, atau juga dikenal sebagai *Data Encryption Algorithm* (DEA) oleh ANSI dan DEA-1 oleh ISO, merupakan algoritma kriptografi yang paling umum digunakan saat ini. Sejarahnya DES dimulai dari permintaan pemerintah Amerika Serikat untuk memasukkan proposal enskripsi. DES memiliki sejarah dari Lucifer, enkripsi yang dikembangan di IBM kala itu. Horst Feistel merupakan salah satu periset yang mula-mula mengembangkan DES ketika bekerja di IBM Watson Laboratory di Yorktown Heights, New York. DES baru secara resmi digunakan oleh pemerintah Amerika Serikat di tahun 1977.

Aplikasi yang menggunakan DES antara lain:

- enkripsi dari password di sistem UNIX
- berbagai aplikasi di bidang perbankan

Memecahkan DES

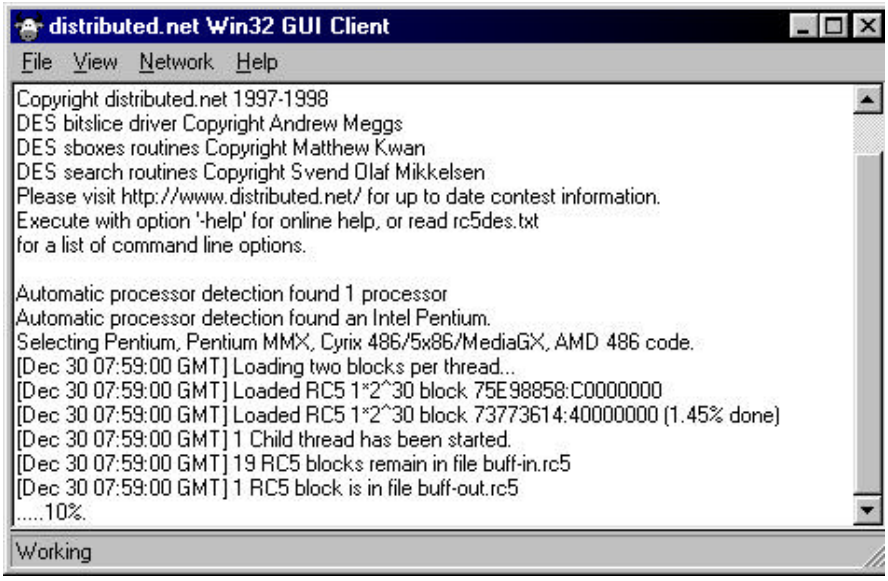
DES merupakan block chiper yang beroperasi dengan menggunakan blok berukuran 64-bit dan kunci berukuran 56-bit. Brute force attack dengan mencoba segala kombinasi membutuhkan 2^{56} kombinasi atau sekitar $7x\ 10^{17}$ atau 70 juta milyar kombinasi.

DES dengan penggunaan yang biasa (*cookbook mode*) dengan panjang kunci 56 bit saat ini sudah dapat dianggap tidak aman karena sudah berhasil dipecahkan dengan metoda coba-coba (*brute force attack*).

Ada berbagai group yang mencoba memecahkan DES dengan berbagai cara. Salah satu group yang bernama ***distributed.net*** menggunakan teknologi Internet untuk memecahkan problem ini menjadi sub-problem yang kecil (dalam ukuran blok). Pengguna dapat menjalankan sebuah program yang khusus dikembangkan oleh tim ini untuk mengambil beberapa blok, via Internet, kemudian memecahkannya di komputer pribadinya. Program yang disediakan meliputi berbagai operating system seperti Windows, DOS, berbagai variasi Unix, Macintosh. Blok yang sudah diproses dikembalikan ke *distributed.net* via Internet. Dengan cara ini puluhan ribu orang, termasuk penulis, membantu memecahkan DES. Mekanisme ini dapat memecahkan DES dalam waktu 30 hari.

Sebuah group lain yang disebut *Electronic Frontier Foundation* (EFF) membuat sebuah komputer yang dilengkapi dengan *Integrated Circuit chip DES cracker*. Dengan mesin seharga US$50.000 ini mereka dapat memecahkan DES 56-bit dalam waktu rata-rata empat (4) sampai lima (5) hari. DES cracker yang mereka kembangkan dapat melakukan eksplorasi keseluruhan dari 56-bit *keyspace* dalam waktu sembilan (9) hari. Dikarenakan 56-bit memiliki 2^{16} (atau 65536) *keyspace* dibandingkan DES dengan 40-bit, maka untuk memecahkan DES 40-bit hanya dibutuhkan waktu sekitar 12 detik[1]. Dikarenakan

hukum average, waktu rata-rata untuk memecahkan DES 40-bit adalah 6 detik.



GAMBAR 2.2. Contoh peragaan client distributed.net untuk Windows 95

Perlu diingat bahwa group seperti EFF merupakan group kecil dengan budget yang terbatas. Dapat dibayangkan sistem yang dimiliki oleh *National Security Agency* (NSA) dari pemerintah Amerika Serikat[1]. Tentunya mereka dapat memecahkan DES dengan lebih cepat.

1. Sembilan hari sama dengan 777.600 detik. Jika angka tersebut dibagi dengan 65.536 maka hasilnya adalah sekitar 12 detik.
1. Budget dari NSA termasuk yang rahasia (*classified*).

Bahan bacaan DES

Banyak sudah buku, artikel yang memuat informasi tentang DES. Bagi anda yang berminat untuk mempelajari DES lebih lanjut, silahkan mengunakan referensi [7, 9, 20, 26 - Chapter 12].

Untuk DES cracker dari EFF, silahkan kunjungi web sitenya di http://www.eff.org/descracker.html

Hash function - integrity checking

Salah satu cara untuk menguji integritas sebuah data adalah dengan memberikan "checksum" atau tanda bahwa data tersebut tidak berubah. Cara yang paling mudah dilakukan adalah dengan menjumlahkan karakter-karakter atau data-data yang ada sehingga apabila terjadi perubahan, hasil penjumlahan menjadi berbeda. Cara ini tentunya mudah dipecahkan dengan menggunakan kombinasi data yang berbeda akan tetapi menghasilkan hasil penjumlahan yang sama.

Pada sistem digital biasanya ada beberapa mekanisme pengujian integritas seperti antara lain:

- parity checking
- checksum
- hash function

Hash function merupakan fungsi yang bersifat satu arah dimana jika kita masukkan data, maka dia akan menghasilkan sebuah "checksum" atau "fingerprint" dari data tersebut. Ada beberapa hash function yang umum digunakan, antara lain:

- MD5
- SHA

Latihan 6. Gunakan MD5 untuk menghasilkan fingerprint dari kalimat berikut: "Saya pesan 10 buah komputer." (tanpa tanda petik). Kemudian nbadingkan hasil MD5 dengan kalimat: "Saya pesan 11 buah komputer."

Contoh latihan di atas dapat dijalankan pada sistem UNIX yang memiliki program "md5" seperti di bawah ini.

```
unix% echo 'Saya pesan 10 buah komputer.' | md5
5F736F18556E3B8D90E50299C7345035
unix% echo 'Saya pesan 11 buah komputer.' | md5
9CB9AD1A369512C96C74236B959780D3
```

Hasil yang serupa dapat dilakukan dengan menggunakan SHA atau algoritma dan program lainnya.

BAB 3

Evaluasi Keamanan Sistem Informasi

"Information is what feeds hacker...
Hacking appeals: it's the control, the adrenaline, the knowledge,
the having what you're not supposed to have."
-- Jon Littman, in "The Fugitive Game: online with Kevin Mitnic"

Apabila anda telah memiliki sebuah sistem informasi, bab ini akan membantu anda untuk mengevaluasi keamanan sistem informasi yang anda miliki.

Meski sebuah sistem informasi sudah dirancang memiliki perangkat pengamanan, dalam operasi masalah keamanan harus selalu dimonitor. Hal ini disebabkan oleh beberapa hal, antara lain:

- Ditemukannya lubang keamanan (*security hole*) yang baru. Perangkat lunak dan perangkat keras biasanya sangat kompleks sehingga tidak mungkin untuk diuji seratus persen. Kadang-kadang ada lubang keamanan yang ditimbulkan oleh kecerobohan implementasi.

- Kesalahan konfigurasi. Kadang-kadang karena lalai atau alpa, konfigurasi sebuah sistem kurang benar sehingga menimbulkan lubang keamanan. Misalnya *mode* (*permission* atau kepemilikan) dari berkas yang menyimpan password (`/etc/passwd` di sistem UNIX) secara tidak sengaja diubah sehingga dapat diubah atau ditulis oleh orang-orang yang tidak berhak.
- Penambahan perangkat baru (hardware dan/atau software) yang menyebabkan menurunnya tingkat security atau berubahnya metoda untuk mengoperasikan sistem. Operator dan administrator harus belajar lagi. Dalam masa belajar ini banyak hal yang jauh dari sempurna, misalnya server atau software masih menggunakan konfigurasi awal dari vendor (dengan password yang sama).

Sumber lubang keamanan

Lubang keamanan (*security hole*) dapat terjadi karena beberapa hal; salah disain (*design flaw*), salah implementasi, salah konfigurasi, dan salah penggunaan.

Salah Disain

Lubang keamanan yang ditimbulkan oleh salah disain umumnya jarang terjadi. Akan tetapi apabila terjadi sangat sulit untuk diperbaiki. Akibat disain yang salah, maka biarpun dia diimplementasikan dengan baik, kelemahan dari sistem akan tetap ada.

Contoh sistem yang lemah disainnya adalah algoritma enkripsi ROT13 atau Caesar cipher, dimana karakter digeser 13 huruf atau 3 huruf. Meskipun diimplementasikan dengan programming yang sangat teliti, siapapun yang mengetahui algoritmanya dapat memecahkan enkripsi tersebut.

Contoh lain lubang keamanan yang dapat dikategorikan kedalam kesalahan disain adalah disain urutan nomor (*sequence numbering*) dari paket TCP/IP. Kesalahan ini dapat dieksploitasi sehingga timbul masalah yang dikenal dengan nama "*IP spoofing*", yaitu sebuah host memalsukan diri seolah-olah menjadi host lain dengan membuat paket palsu setelah mengamati urutan paket dari host yang hendak diserang. Bahkan dengan mengamati cara mengurutkan nomor packet bisa dikenali sistem yang digunakan. Mekanisme ini digunakan oleh program *nmap* dan *queso* untuk mendeteksi *operating system* (OS) dari sebuah sistem, yang disebut *fingerprinting*. Contoh dan informasi yang lebih lengkap mengenai masalah kelemahan protokol TCP/IP dapat dilihat pada referensi [1].

Implementasi kurang baik

Lubang keamanan yang disebabkan oleh kesalahan implementasi sering terjadi. Banyak program yang diimplementasikan secara terburu-buru sehingga kurang cermat dalam pengkodean. Akibatnya cek atau testing yang harus dilakukan menjadi tidak dilakukan. Sebagai contoh, seringkali batas ("*bound*") dari sebuah "*array*" tidak dicek sehingga terjadi yang disebut *out-of-bound array* atau *buffer overflow* yang dapat dieksploitasi (misalnya overwrite ke variable berikutnya). Lubang keamanan yang terjadi karena masalah ini sudah sangat banyak, dan yang mengherankan terus terjadi, seolah-olah para programmer tidak belajar dari pengalaman[1].

Contoh lain sumber lubang keamanan yang disebabkan oleh kurang baiknya implementasi adalah kealpaan memfilter karakter-karakter yang aneh-aneh yang dimasukkan sebagai input dari sebuah program (misalnya input dari *CGI-script*[2]) sehingga sang program

1. Memang kesalahan tidak semata-mata ditimpakan kepada pembuat program karena seringkali mereka dikejar deadline oleh management tingkat atas untuk merilis soft-warenya.
2. Tentang CGI-script akan dijelaskan di bagian lain.

dapat mengakses berkas atau informasi yang semestinya tidak boleh diakses.

Salah konfigurasi

Meskipun program sudah diimplementasikan dengan baik, masih dapat terjadi lubang keamanan karena salah konfigurasi. Contoh masalah yang disebabkan oleh salah konfigurasi adalah berkas yang semestinya tidak dapat diubah oleh pemakai secara tidak sengaja menjadi "*writeable*". Apabila berkas tersebut merupakan berkas yang penting, seperti berkas yang digunakan untuk menyimpan password, maka efeknya menjadi lubang keamanan. Kadangkala sebuah komputer dijual dengan konfigurasi yang sangat lemah. Ada masanya workstation Unix di perguruan tinggi didistribusikan dengan berkas `/etc/aliases` (berguna untuk mengarahkan e-mail), `/etc/utmp` (berguna untuk mencatat siapa saja yang sedang menggunakan sistem) yang dapat diubah oleh siapa saja. Contoh lain dari salah konfigurasi adalah adanya program yang secara tidak sengaja diset menjadi "*setuid root*" sehingga ketika dijalankan pemakai memiliki akses seperti *super user* (*root*) yang dapat melakukan apa saja.

Salah menggunakan program atau sistem

Salah penggunaan program dapat juga mengakibatkan terjadinya lubang keamanan. Kesalahan menggunakan program yang dijalankan dengan menggunakan account root (super user) dapat berakibat fatal. Sering terjadi cerita horor dari sistem administrator baru yang teledor dalam menjalankan perintah "`rm -rf`" di sistem UNIX (yang menghapus berkas atau direktori beserta sub direktori di dalamnya). Akibatnya seluruh berkas di sistem menjadi hilang mengakibatkan *Denial of Service* (DoS). Apabila sistem yang digunakan ini digunakan bersama-sama, maka akibatnya dapat lebih fatal lagi. Untuk itu perlu berhati-hati dalam menjalan program, terutama apabila dilakukan dengan menggunakan account administrator seperti *root* tersebut.

Kesalahan yang sama juga sering terjadi di sistem yang berbasis MS-DOS. Karena sudah mengantuk, misalnya, ingin melihat daftar berkas di sebuah direktori dengan memberikan perintah "dir *.*" ternyata salah memberikan perintah menjadi "`del *.*`" (yang juga menghapus seluruh file di direktori tersebut).

Penguji keamanan sistem

Dikarenakan banyaknya hal yang harus dimonitor, administrator dari sistem informasi membutuhkan "*automated tools*", perangkat pembantu otomatis, yang dapat membantu menguji atau meng-evaluasi keamanan sistem yang dikelola. Untuk sistem yang berbasis UNIX ada beberapa tools yang dapat digunakan, antara lain:

- *Cops*
- *Tripwire*
- *Satan/Saint*
- *SBScan*: localhost security scanner

Untuk sistem yang berbasis Windows NT ada juga program semacam, misalnya program *Ballista* yang dapat diperoleh dari: <http://www.secnet.com>

Selain program-program (tools) yang terpadu (*integrated*) seperti yang terdapat pada daftar di atas, ada banyak program yang dibuat oleh hackers untuk melakukan "coba-coba". Program-program seperti ini, yang cepat sekali bermunculuan, biasanya dapat diperoleh (download) dari Internet melalui tempat-tempat yang berhubungan dengan keamanan, seperti misalnya "*Rootshell*". (Lihat "Sumber informasi dan organisasi yang berhubungan dengan keamanan sistem informasi" on page 83.) Contoh program coba-coba ini antara lain:

- *crack*: program untuk menduga atau memecahkan password dengan menggunakan sebuah atau beberapa kamus (*dictionary*). Program crack ini melakukan brute force cracking dengan mencoba mengenkripsikan sebuah kata yang diambil dari kamus, dan kemudian membandingkan hasil enkripsi dengan password yang ingin dipecahkan. Bila belum sesuai, maka ia akan mengambil kata selanjutnya, mengenkripsikan, dan membandingkan kembali. Hal ini dijalankan terus menerus sampai semua kata di kamus dicoba. Selain menggunakan kata langsung dari kamus, crack juga memiliki program heuristic dimana bolak balik kata (dan beberapa modifikasi lain) juga dicoba. Jadi, jangan sekali-kali menggunakan password yang terdapat dalam kamus (bahasa apapun).
- *land* dan *latierra*: program yang dapat membuat sistem Windows 95/NT menjadi macet (*hang, lock up*). Program ini mengirimkan sebuah paket yang sudah di"*spoofed*" sehingga seolah-olah paket tersebut berasal dari mesin yang sama dengan menggunakan port yang terbuka (misalnya port 113 atau 139).
- *ping-o-death*: sebuah program (*ping*) yang dapat meng-crash-kan Windows 95/NT dan beberapa versi Unix.
- *winuke*: program untuk memacetkan sistem berbasis Windows

Probing Services

Servis di Internet umumnya dilakukan dengan menggunakan protokol TCP atau UDP. Setiap servis dijalankan dengan menggunakan port yang berbeda, misalnya:

- SMTP, untuk mengirim dan menerima e-mail, TCP, port 25
- POP3, untuk mengambil e-mail, TCP, port 110

Contoh di atas hanya sebagian dari servis yang tersedia. Di sistem UNIX, lihat berkas `/etc/services` dan `/etc/inetd.conf` untuk melihat servis apa saja yang dijalankan oleh server atau komputer yang bersangkutan.

Pemilihan servis apa saja tergantung kepada kebutuhan dan tingkat keamanan yang diinginkan. Sayangnya seringkali sistem yang dibeli atau dirakit menjalankan beberapa servis utama sebagai "default". Kadang-kadang beberapa servis harus dimatikan karena ada kemungkinan dapat dieksploitasi oleh cracker. Untuk itu ada beberapa program yang dapat digunakan untuk melakukan "*probe*" (meraba) servis apa saja yang tersedia. Program ini juga dapat digunakan oleh kriminal untuk melihat servis apa saja yang tersedia di sistem yang akan diserang dan berdasarkan data-data yang diperoleh dapat melancarkan serangan.

Untuk beberapa servis yang berbasis TCP/IP, proses probe dapat dilakukan dengan menggunakan program telnet. Misalnya untuk melihat apakah ada servis e-mail dengan menggunakan SMTP digunakan telnet ke port 25.

```
unix% telnet target.host.com 25
Trying 127.0.0.1...
Connected to target.host.com.
Escape character is '^]'.
220 dma-baru ESMTP Sendmail 8.9.0/8.8.5; Mon, 22 Jun 1998
10:18:54 +0700
```

Dalam contoh di atas terlihat bahwa ada servis SMTP di server tersebut dengan menggunakan program *Sendmail* versi 8.9.0. Adanya informasi tentang sistem yang digunakan ini sebetulnya sangat tidak disarankan karena dengan mudah orang dapat mengetahui kebocoran sistem (jika software dengan versi tersebut memiliki lubang keamanan).

Untuk servis lain, seperti POP atau POP3 dapat dilakukan dengan cara yang sama dengan menggunakan nomor "port" yang sesuai dengan servis yang diamati.

```
unix% telnet localhost 110
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
+OK QPOP (version 2.2) at dma-baru.paume.itb.ac.id starting.
+<20651.898485542@dma-baru.paume.itb.ac.id>
quit
```

```
+OK Pop server at dma-baru.paume.itb.ac.id signing off.
Connection closed by foreign host.
```

Latihan 7. Lakukan probing ke sebuah POP server. Gunakan POP server yang dipersiapkan khusus untuk latihan ini. Jangan lakukan probing ke server milik orang lain tanpa ijin.

Proses probing tersebut dapat dilakukan secara otomatis, sehinga menguji semua port yang ada, dengan menggunakan beberapa program paket seperti didaftarkan di bawah ini.

Paket probe untuk sistem UNIX

- *nmap*
- *strobe*
- *tcpprobe*

Latihan 8. Gunakan nmap, strobe, atau tcpprobe untuk melakukan probe terhadap sebuah server yang sudah dipersiapkan untuk latihan ini. Jangan melakukan probe ke server milik orang lain tanpa ijin.

Untuk melakukan probing ke sistem dengan nomor IP 192.168.1.1 dengan menggunakan program strobe:

```
unix% strobe 192.168.1.1
unix% strobe 192.168.1.1 -b 1 -e 80
```

Untuk melakukan probing apakah komputer dengan range nomor IP 192.168.1.1 sampai dengan 192.168.1.10 memiliki FTP server (port 21) dapat dilakukan dengan menggunakan nmap dengan perintah di bawah ini:

```
unix% nmap 192.168.1.1-10 -p 21
```

Probe untuk sistem Window 95/98/NT

- *NetLab*
- *Cyberkit*
- *Ogre*

Apabila anda seorang sistem administrator, anda dapat memasang program yang memonitor adanya probing ke sistem yang anda kelola. Probing biasanya meninggalkan jejak di berkas log di sistem anda. Dengan mengamati entry di dalam berkas log dapat diketahui adanya probing.

```
root# tail /var/log/syslog
May 16 15:40:42 epson tcplogd: "Syn probe"
notebook[192.168.1.4]:[8422]->epson[192.168.1.2]:[635]
May 16 15:40:42 epson tcplogd: "Syn probe"
notebook[192.168.1.4]:[8423]->epson[192.168.1.2]:ssl-ldap
May 16 15:40:42 epson tcplogd: "Syn probe"
notebook[192.168.1.4]:[8426]->epson[192.168.1.2]:[637]
May 16 15:40:42 epson tcplogd: "Syn probe"
notebook[192.168.1.4]:[8429]->epson[192.168.1.2]:[638]
May 16 15:40:43 epson tcplogd: "Syn probe"
notebook[192.168.1.4]:[8430]->epson[192.168.1.2]:[639]
May 16 15:40:43 epson tcplogd: "Syn probe"
notebook[192.168.1.4]:[8437]->epson[192.168.1.2]:[640]
May 16 15:40:43 epson tcplogd: "Syn probe"
notebook[192.168.1.4]:[8441]->epson[192.168.1.2]:[641]
May 16 15:40:43 epson tcplogd: "Syn probe"
notebook[192.168.1.4]:[8445]->epson[192.168.1.2]:[642]
May 16 15:40:43 epson tcplogd: "Syn probe"
notebook[192.168.1.4]:[8454]->epson[192.168.1.2]:[643]
```

Contoh di atas menunjukkan *entry* di berkas *syslog* dimana terjadi probing dari komputer yang di beri nama *notebook* dengan nomor IP 192.168.1.4.

Selain itu, ada juga program untuk memonitor probe seperti paket program *courtney, portsentry* dan *tcplogd.*

OS fingerprinting

Mengetahui *operating system* (OS) dari target yang akan diserang merupakan salah satu pekerjaan yang dilakukan oleh seorang cracker. Setelah mengetahui OS yang dituju, dia dapat melihat database kelemahan sistem yang dituju. *Fingerprinting* merupakan istilah yang umum digunakan untuk menganalisa OS sistem yang dituju [10].

Fingerprinting dapat dilakukan dengan berbagai cara. Cara yang paling konvensional adalah melakukan telnet ke server yang dituju. Jika server tersebut kebetulan menyediakan servis telnet, seringkali ada banner yang menunjukkan nama OS beserta versinya.

```
unix% telnet 192.168.1.4
Trying 192.168.1.4...
Connected to 192.168.1.4.
Escape character is '^]'.
Linux 2.0.33 (rock.pau-mikro.org) (ttyp0)
login:
```

Apabila sistem tersebut tidak menyediakan servis telnet akan tetapi menyediakan servis FTP, maka informasi juga sering tersedia. Servis FTP tersedia di port 21. Dengan melakukan telnet ke port tersebut dan memberikan perintah “SYST” anda dapat mengetahui versi dari OS yang digunakan seperti contoh di bawah ini.

```
unix% telnet ftp.netscape.com 21
Trying 207.200.74.26...
Connected to ftp.netscape.com.
Escape character is '^]'.
220 ftp29 FTP server (UNIX(r) System V Release 4.0) ready.
SYST
215 UNIX Type: L8 Version: SUNOS
```

Jika server tersebut tidak memiliki FTP server akan tetapi menjalankan Web server, masih ada cara untuk mengetahui OS yang digunakan dengan menggunakan program *netcat* (*nc*) seperti contoh di bawah ini (dimana terlihat OS yang digunakan adalah Debian GNU):

```
$ echo -e "GET / HTTP/1.0\n\n" | nc localhost 80 | \
```

```
grep "^Server:"
Server: Apache/1.3.3 (Unix) Debian/GNU
```

Cara fingerprinting yang lebih canggih adalah dengan menganalisa respon sistem terhadap permintaan (request) tertentu. Misalnya dengan menganalisa nomor urut packet TCP/IP yang dikeluarkan oleh server tersebut dapat dipersempit ruang jenis dari OS yang digunakan.

Ada beberapa tools untuk melakukan deteksi OS ini antara lain:

- *nmap*
- *queso*

Berikut ini adalah contoh penggunaan program *queso* untuk mendeteksi OS dari sistem yang menggunakan nomor IP 192.168.1.1. Kebetulan sistem ini adalah sistem Windows 95.

```
unix# queso 192.168.1.1
192.168.1.1:80  * Not Listen, Windoze 95/98/NT
```

Penggunaan program penyerang

Salah satu cara untuk mengetahui kelemahan sistem informasi anda adalah dengan menyerang diri sendiri dengan paket-paket program penyerang (*attack*) yang dapat diperoleh di Internet. Dengan menggunakan program ini anda dapat mengetahui apakah sistem anda rentan dan dapat dieksploitasi oleh orang lain. Perlu diingat bahwa **jangan menggunakan program-program tersebut untuk menyerang sistem lain** (sistem yang tidak anda kelola). Ini tidak etis dan anda dapat diseret ke pengadilan. Beberapa program penyerangan dicontohkan di Bab “Eksploitasi Keamananan” on page 69.

Selain program penyerang yang sifatnya agresif melumpuhkan sistem yang dituju, ada juga program penyerang yang sifatnya melakukan pencurian atau penyadapan data. Untuk penyadapan

data, biasanya dikenal dengan istilah "*sniffer*". Meskipun data tidak dicuri secara fisik (dalam artian menjadi hilang), sniffer ini sangat berbahaya karena dia dapat digunakan untuk menyadap password dan informasi yang sensitif. Ini merupakan serangan terhadap aspek privacy.

Contoh program penyadap (*sniffer*) antara lain:

- *pcapture* (Unix)
- *sniffit* (Unix)
- *tcpdump* (Unix)
- *WebXRay* (Windows)

Penggunaan sistem pemantau jaringan

Sistem pemantau jaringan (*network monitoring*) dapat digunakan untuk mengetahui adanya lubang keamaman. Misalnya apabila anda memiliki sebuah server yang semetinya hanya dapat diakses oleh orang dari dalam, akan tetapi dari pemantau jaringan dapat terlihat bahwa ada yang mencoba mengakses melalui tempat lain. Selain itu dengan pemantau jaringan dapat juga dilihat usaha-usaha untuk melumpuhkan sistem dengan melalui *denial of service attack* (DoS) dengan mengirimkan packet yang jumlahnya berlebihan.

Network monitoring biasanya dilakukan dengan menggunakan protokol SNMP *(Simple Network Management Protoco*l) [7]. Pada saat buku ini ditulis, SNMP versi 1 yang paling banyak digunakan meskipun SNMP versi 2 sudah keluar. Sayangnya, tingkat keamanan dari SMNP versi 1 sangat rendah sehingga memungkinkan penyadapan oleh orang yang tidak berhak

Contoh-contoh program network monitoring / management antara lain:

- *Etherboy* (Windows), *Etherman* (Unix)
- HP *Openview* (Windows)

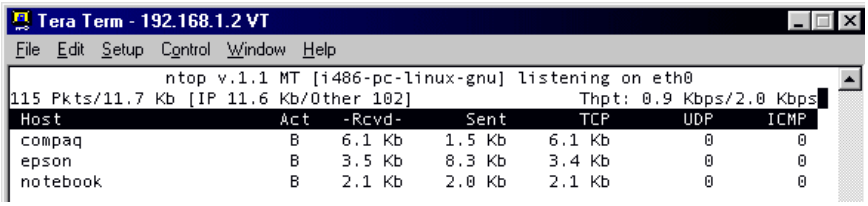
- *Packetboy* (Windows), *Packetman* (Unix)
- SNMP Collector (Windows)
- *Webboy* (Windows)

Contoh program pemanatu jaringan yang tidak menggunakan SNMP antara lain:

- *iplog*, *icmplog*, *updlog*, yang merupakan bagian dari paket *iplog* untuk memantau paket IP, ICMP, UDP.
- *iptraf*, sudah termasuk dalam paket Linux Debian *netdiag*
- *netwatch*, sudah termasuk dalam paket Linux Debian *netdiag*
- *ntop*, memantau jaringan seperti program *top* yang memantau proses di sistem Unix (lihat contoh gambar tampilannya)
- *trafshow*, menunjukkan traffic antar hosts dalam bentuk text-mode

Contoh peragaan *trafshow* di sebuah komputer yang bernama epson, dimana ditunjukkan sesi *ssh* (dari komputer compaq) dan *ftp* (dari komputer notebook).

```
epson (traffic) 0 days 00 hrs 00 min 46 sec
tcp  epson.insan.co.id   ssh    compaq 558 3096        832
tcp  epson.insan.co.id   ftp    notebook 1054 422       381
9K total, 0K bad, 0K nonip - 9K tcp, 0K udp, 0K icmp, 0K unkn
```



GAMBAR 3.1. Contoh tampilan ntop

BAB 4

Mengamankan Sistem Informasi

"if a hacker obtains a login on a machine,
there is a good chance he can become root sooner or later."
-- Bill Cheswick, in "An evening with Berferd:
in which a cracker is lured, endured, and studied")

Dalam bab sebelumnya telah dibahas cara-cara untuk mengevaluasi sistem anda. Maka bab ini akan membahas cara-cara untuk mengamankan sistem informasi anda.

Pada umunya, pengamanan dapat dikategorikan menjadi dua jenis: pencegahan (*preventif*) dan pengobatan (*recovery*). Usaha pencegahan dilakukan agar sistem informasi tidak memiliki lubang keamanan, sementara usaha-usaha pengobatan dilakukan apabila lubang keamanan sudah dieksploitasi.

Pengamanan sistem informasi dapat dilakukan melalui beberapa layer yang berbeda. Misalnya di layer "transport", dapat digunakan "*Secure Socket Layer*" (SSL). Metoda ini misalnya umum digunakan untuk Web Site. Secara fisik, sistem anda dapat juga diamankan

dengan menggunakan “firewall” yang memisahkan sistem anda dengan Internet. Penggunaan teknik enkripsi dapat dilakukan di tingkat aplikasi sehingga data-data anda atau e-mail anda tidak dapat dibaca oleh orang yang tidak berhak.

Mengatur akses (Access Control)

Salah satu cara yang umum digunakan untuk mengamankan informasi adalah dengan mengatur akses ke informasi melalui mekanisme “*access control*”. Implementasi dari mekanisme ini antara lain dengan menggunakan “*password*”.

Di sistem UNIX, untuk menggunakan sebuah sistem atau komputer, pemakai diharuskan melalui proses *authentication* dengan menuliskan “*userid*” dan “*password*”. Informasi yang diberikan ini dibandingkan dengan userid dan password yang berada di sistem. Apabila keduanya valid, pemakai yang bersangkutan diperbolehkan menggunakan sistem. Apabila ada yang salah, pemakai tidak dapat menggunakan sistem. Informasi tentang kesalahan ini biasanya dicatat dalam berkas *log*. Besarnya informasi yang dicatat bergantung kepada konfigurasi dari sistem setempat. Misalnya, ada yang menuliskan informasi apabila pemakai memasukkan *userid* dan *password* yang salah sebanyak tiga kali. Ada juga yang langsung menuliskan informasi ke dalam berkas *log* meskipun baru satu kali salah. Informasi tentang waktu kejadian juga dicatat. Selain itu asal hubungan (connection) juga dicatat sehingga administrator dapat memeriksa keabsahan hubungan.

Password di sistem UNIX

Akses ke sistem UNIX menggunakan password yang biasanya disimpan di dalam berkas `/etc/passwd`. Di dalam berkas ini disimpan nama, userid, password, dan informasi-informasi lain yang digunakan oleh bermacam-macam program. Contoh isi berkas password dapat dilihat di bawah ini.

```
root:fi3sED95ibqR7:0:1:System Operator:/:/sbin/sh
daemon:*:1:1::/tmp:
rahard:d98skjhj9l:72:98:Budi Rahardjo:/home/rahard:/bin/csh
```

TABLE 2. Penjelasan contoh isi berkas password

Field	Isi
`rahard`	Nama atau userid pemakai
`d98skjhj9l`	password yang sudah terenkripsi (*encrypted password*)
`72`	UID, user identification number
`98`	GID, group identification number
`Budi Rahardjo`	Nama lengkap dari pemakai (sering juga disebut GECOS[a] atau GCOS field)
`/home/rahard`	home directory dari pemakai
`/bin/csh`	shell dari pemakai

a. GECOS = General Electric Computer Operating System. Di masa lalu, pemakai juga memiliki account di komputer yang lebih besar, yaitu komputer GECOS. Informasi ini disimpan dalam berkas ini untuk memudahkan batch job yang dijalankan melalui sebuah Remote Job Entry. [12]

Pada sistem UNIX lama, biasanya berkas `/etc/passwd` ini "readable", yaitu dapat dibaca oleh siapa saja. Meskipun kolom password di dalam berkas itu berisi "*encrypted password*" (password yang sudah terenkripsi), akan tetapi ini merupakan potensi sumber lubang keamanan. Seorang pemakai yang nakal, dapat mengambil berkas ini (karena "*readable*"), misalnya men-download berkas ini ke komputer di rumahnya, atau mengirimkan berkas ini kepada kawannya. Ada program tertentu yang dapat digunakan untuk memecah password tersebut. Contoh program ini antara lain: *crack* (UNIX), *viper* (perl script), dan *cracker jack* (DOS).

Program "*password cracker*" ini tidak dapat mencari tahu kata kunci dari kata yang sudah terenkripsi. Akan tetapi, yang dilakukan oleh program ini adalah melakukan coba-coba (*brute force attack*). Salah satu caranya adalah mengambil kata dari kamus (*dictionary*) kemudian mengenkripsinya. Apabila hasil enkripsi tersebut sama dengan password yang sudah terenkripsi (*encrypted password*), maka

kunci atau passwordnya ketemu. Selain melakukan "*lookup*" dengan menggunakan kamus, biasanya program "*password cracker*" tersebut memiliki beberapa algoritma *heuristic* seperti menambahkan angka di belakangnya, atau membaca dari belakang (terbalik), dan seterusnya. Inilah sebabnya jangan menggunakan password yang terdapat dalam kamus, atau kata-kata yang umum digunakan (seperti misalnya nama kota atau lokasi terkenal).

Shadow Password

Salah satu cara untuk mempersulit pengacau untuk mendapatkan berkas yang berisi password (meskipun terenkripsi) adalah dengan menggunakan "*shadow password*". Mekanisme ini menggunakan berkas `/etc/shadow` untuk menyimpan encrypted password, sementara kolom password di berkas `/etc/passwd` berisi karakter "x". Berkas `/etc/shadow` tidak dapat dibaca secara langsung oleh pemakai biasa.

Latihan 9. Perhatikan sistem UNIX anda. Apakah sistem itu menggunakan fasilitas shadow password atau tidak?

Memilih password

Dengan adanya kemungkinan password ditebak, misalnya dengan menggunakan program password cracker, maka memilih password memerlukan perhatian khusus. Berikut ini adalah daftar hal-hal yang sebaiknya tidak digunakan sebagai password.

- Nama anda, nama istri / suami anda, nama anak, ataupun nama kawan.
- Nama komputer yang anda gunakan.
- Nomor telepon atau plat nomor kendaran anda.
- Tanggal lahir.
- Alamat rumah.
- Nama tempat yang terkenal.
- Kata-kata yang terdapat dalam kamus (bahasa Indonesia maupun bahasa Inggris).

- Password dengan karakter yang sama diulang-ulang.
- Hal-hal di atas ditambah satu angka.

Menutup servis yang tidak digunakan

Seringkali sistem (perangkat keras dan/atau perangkat lunak) diberikan dengan beberapa servis dijalankan sebagai *default*. Sebagai contoh, pada sistem UNIX servis-servis berikut sering dipasang dari vendornya: *finger*, *telnet*, *ftp*, *smtp*, *pop*, *echo*, dan seterusnya. Servis tersebut tidak semuanya dibutuhkan. Untuk mengamankan sistem, servis yang tidak diperlukan di server (komputer) tersebut sebaiknya dimatikan. Sudah banyak kasus yang menunjukkan *abuse* dari servis tersebut, atau ada lubang keamanan dalam servis tersebut akan tetapi sang administrator tidak menyadari bahwa servis tersebut dijalankan di komputernya.

Latihan 10. Periksa sistem UNIX anda, servis apa saja yang dijalankan di sana? Dari mana anda tahu servis-servis yang dijalankan?

Servis-servis di sistem UNIX ada yang dijalankan dari "*inetd*" dan ada yang dijalankan sebagai *daemon*. Untuk mematikan servis yang dijalankan dengan menggunakan fasilitas *inet*, periksa berkas `/etc/inetd.conf`, matikan servis yang tidak digunakan (dengan memberikan tanda komentar #) dan memberitahu *inetd* untuk membaca berkas konfigurasinya (dengan memberikan signal HUP kepada PID dari proses *inetd*).

```
unix# ps -aux | grep inetd
105 inetd
unix# kill -HUP 105
```

Untuk sistem Solaris atau yang berbasis System V, gunakan perintah "`ps -eaf`" sebagai pengganti perintah "`ps -aux`". Lebih jelasnya silahkan baca manual dari perintah *ps*.

Untuk servis yang dijalankan sebagai *daemon* dan dijalankan pada waktu *startup* (*boot*), perhatikan skrip boot dari sistem anda.

- SunOS: /etc/rc.*
- Linux Debian: /etc/init.d/*

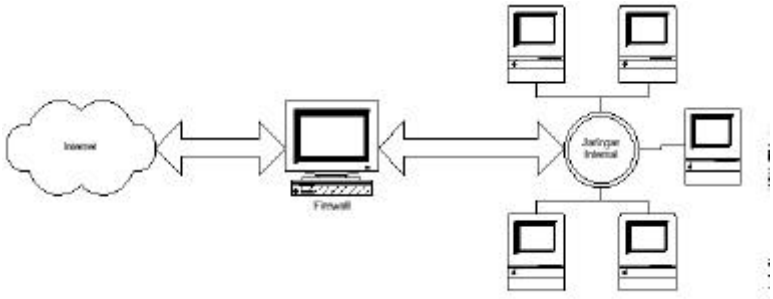
Memasang Proteksi

Untuk lebih meningkatkan keamanan sistem informasi, proteksi dapat ditambahkan. Proteksi ini dapat berupa filter (secara umum) dan yang lebih spesifik adalah firewall. Filter dapat digunakan untuk memfilter e-mail, informasi, akses, atau bahkan dalam level packet. Sebagai contoh, di sistem UNIX ada paket program "*tcpwrapper*" yang dapat digunakan untuk membatasi akses kepada servis atau aplikasi tertentu. Misalnya, servis untuk "*telnet*" dapat dibatasi untuk untuk sistem yang memiliki nomor IP tertentu, atau memiliki domain tertentu. Sementara firewall dapat digunakan untuk melakukan filter secara umum.

Untuk mengetahui apakah server anda menggunakan *tcpwrapper* atau tidak, periksa isi berkas `/etc/inetd.conf`. Biasanya *tcpwrapper* dirakit menjadi "`tcpd`". Apabila servis di server anda (misalnya *telnet* atau *ftp*) dijalankan melalui `tcpd`, maka server anda menggunakan *tcpwrapper*. Biasanya, konfigurasi *tcpwrapper* (`tcpd`) diletakkan di berkas `/etc/hosts.allow` dan `/etc/hosts.deny`.

Firewall

Firewall merupakan sebuah perangkat yang diletakkan antara Internet dengan jaringan internal (Lihat Figure 4.1 on page 55). Informasi yang keluar atau masuk harus melalui firewall ini.



GAMBAR 4.1. Contoh sebuah Firewall

Tujuan utama dari firewall adalah untuk menjaga (*prevent*) agar akses (ke dalam maupun ke luar) dari orang yang tidak berwenang (*unauthorized access*) tidak dapat dilakukan. Konfigurasi dari firewall bergantung kepada kebijaksanaan (*policy*) dari organisasi yang bersangkutan, yang dapat dibagi menjadi dua jenis:

- apa-apa yang tidak diperbolehkan secara eksplisit dianggap tidak diperbolehkan (*prohibitted*)
- apa-apa yang tidak dilarang secara eksplisit dianggap diperbolehkan (*permitted*)

Firewall bekerja dengan mengamati paket IP (Internet Protocol) yang melewatinya. Berdasarkan konfigurasi dari firewall maka akses dapat diatur berdasarkan IP address, port, dan arah informasi. Detail dari konfigurasi bergantung kepada masing-masing firewall.

Firewall dapat berupa sebuah perangkat keras yang sudah dilengkapi dengan perangkat lunak tertentu, sehingga pemakai (administrator) tinggal melakukan konfigurasi dari firewall tersebut. Firewall juga dapat berupa perangkat lunak yang ditambahkan kepada sebuah server (baik UNIX maupun Windows NT), yang dikonfigurasi menjadi firewall. Dalam hal ini, sebetulnya perangkat komputer dengan prosesor Intel 80486 sudah cukup untuk menjadi firewall yang sederhana.

Firewall biasanya melakukan dua fungsi; fungsi (IP) filtering dan fungsi proxy. Keduanya dapat dilakukan pada sebuah perangkat komputer (device) atau dilakukan secara terpisah.

Beberapa perangkat lunak berbasis UNIX yang dapat digunakan untuk melakukan IP filtering antara lain:

- *ipfwadm*: merupakan standar dari sistem Linux yang dapat diaktifkan pada level kernel
- *ipchains*: versi baru dari Linux kernel packet filtering yang diharapkan dapat menggantikan fungsi ipfwadm

Fungsi proxy dapat dilakukan oleh berbagai software tergantung kepada jenis proxy yang dibutuhkan, misalnya web proxy, rlogin proxy, ftp proxy dan seterusnya. Di sisi client sering kalai dibutuhkan software tertentu agar dapat menggunakan proxy server ini, seperti misalnya dengan menggunakan SOCKS. Beberapa perangkat lunak berbasis UNIX untuk proxy antara lain:

- *Socks:* proxy server oleh NEC Network Systems Labs
- *Squid*: web proxy server

Informasi mengenai firewall secara lebih lengkap dapat dibaca pada referensi [19, 24] atau untuk sistem Linux dapat dilakukan dengan mengunjungi web site berikut: <http://www.gnatbox.com>.

Pemantau adanya serangan

Sistem pemantau (*monitoring system*) digunakan untuk mengetahui adanya tamu tak diundang (*intruder*) atau adanya serangan (*attack*). Nama lain dari sistem ini adalah "*intruder detection system*" (IDS). Sistem ini dapat memberitahu administrator melalui e-mail maupun melalui mekanisme lain seperti melalui pager.

Ada berbagai cara untuk memantau adanya intruder. Ada yang sifatnya aktif dan pasif. IDS cara yang pasif misalnya dengan memonitor logfile. Contoh software IDS antara lain:

- *Autobuse*, mendeteksi probing dengan memonitor logfile.
- *Courtney*, mendeteksi probing dengan memonitor packet yang lalu lalang
- *Shadow* dari SANS

Pemantau integritas sistem

Pemantau integritas sistem dijalankan secara berkala untuk menguji integratitas sistem. Salah satu contoh program yang umum digunakan di sistem UNIX adalah program *Tripwire*. Program paket *Tripwire* dapat digunakan untuk memantau adanya perubahan pada berkas. Pada mulanya, tripwire dijalankan dan membuat database mengenai berkas-berkas atau direktori yang ingin kita amati beserta "*signature*" dari berkas tersebut. Signature berisi informasi mengenai besarnya berkas, kapan dibuatnya, pemiliknya, hasil *checksum* atau *hash* (misalnya dengan menggunakan program MD5), dan sebagainya. Apabila ada perubahan pada berkas tersebut, maka keluaran dari *hash function* akan berbeda dengan yang ada di database sehingga ketahuan adanya perubahan.

Audit: Mengamati Berkas Log

Segala (sebagian besar) kegiatan penggunaan sistem dapat dicatat dalam berkas yang biasanya disebut "*logfile*" atau "*log*" saja. Berkas log ini sangat berguna untuk mengamati penyimpangan yang terjadi. Kegagalan untuk masuk ke sistem (login), misalnya, tersimpan di dalam berkas log. Untuk itu para administrator diwajibkan untuk rajin memelihara dan menganalisa berkas log yang dimilikinya.

Letak dan isi dari berkas log bergantung kepada operating system yang digunakan. Di sistem berbasis UNIX, biasanya berkas ini berada di direktori `/var/adm` atau `/var/log`. Contoh berkas log

yang ada di sistem Linux Debian dapat dilihat pada Table 3 on page 58.

.

TABLE 3. Berkas Log di sistem Debian Linux

Nama Berkas	Keterangan
/var/adm/auth.log	Berisi informasi yang berhubungan dengan authentication. Gagal login, misalnya, dicatat pada berkas ini.
/var/adm/daemon.log	Informasi mengenai program-program daemon seperti BIND, Sendmail, dsb.
/var/adm/mail.log	Berisi informasi tentang e-mail yang dikirimkan dan diterima serta akses ke sistem email melalui POP dan IMAP.
/var/adm/syslog	Berisi pesan yang dihasilkan oleh program syslog. Kegagalan login tercatat di sini.

Sebagai contoh, berikut ini adalah cuplikan baris isi dari berkas `/var/adm/auth.log`:

```
Apr  8 08:47:12 xact passwd[8518]: password for `inet' changed
by root
Apr  8 10:02:14 xact su: (to root) budi on /dev/ttyp3
```

Baris pertama menunjukkan bawah password untuk pemakai “inet” telah diganti oleh “root”. Baris kedua menunjukkan bahwa pemakai (*user*) yang bernama “budi” melakukan perintah “`su`” (*substitute user*) dan menjadi user “root” (*super user*). Kedua contoh di atas menunjukkan entry yang nampaknya normal, tidak mengandung security hole, dengan asumsi pada baris kedua memang pemakai “budi” diperbolehkan menjadi root. Contoh entry yang agak mencurigakan adalah sebagai berikut.

```
Apr  5 17:20:10 alliance wu-ftpd[12037]: failed login from
ws170.library.msstate.edu [130.18.249.170], m1
Apr  9 18:41:47 alliance login[12861]: invalid password for
`budi' on `ttyp0' from `ppp15.isp.net.id'
```

Baris di atas menunjukkan kegagalan untuk masuk ke sistem melalui fasilitas FTP (baris pertama) dan `telnet` (baris kedua). Pada baris kedua terlihat bahwa user "budi" (atau yang mengaku sebagai user "budi") mencoba masuk melalui login dan gagal memberikan password yang valid. Hal ini bisa terjadi karena ketidak sengajaan, salah memasukkan password, atau bisa juga karena sengaja ingin mencoba-coba masuk dengan userid "budi" dengan password coba-coba. Cara coba-coba ini sering dilakukan dengan mengamati nama user yang berada di sistem tersebut (misalnya dengan menggunakan program `finger` untuk mengetahui keberadaan sebuah user).

Contoh berikut diambil dari isi berkas `/var/adm/mail.log`, yang berfungsi untuk mencatat aktivitas yang berhubungan dengan sistem mail.

```
Apr  9 18:40:31 mx1 imapd[12859]: Login faiure
user=^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P
^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^
P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P
host=vhost.txg.wownet.net
Apr  9 18:40:32 mx1 imapd[12859]: Success, while reading line
user=^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P
^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^
P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P
host=vhost.txg.wownet.net
```

Contoh di atas menunjukkan hal yang sedikit aneh dari akses ke servis email melalui IMAP (ditunjukkan dengan kata "`imapd`" yang merupakan server dari servis IMAP). Pertama, user yang digunakan tidak valid. Kedua, kebetulan administrator tidak memiliki remote user yang berasal dari host yang disebut di atas. Setelah diselidiki, ternyata memang ada lubang keamanan dari implementasi "`imapd`" yang digunakan. Ini diketahui setelah melihat informasi yang ada di web site *CERT* (See "Sumber informasi dan organisasi yang berhubungan dengan keamanan sistem informasi" on page 83.). Untuk itu administrator cepat-cepat menutup servis *imap* tersebut, mengambil dan memasang versi baru dari `imapd` yang tidak memiliki lubang keamanan tersebut.

Contoh-contoh di atas hanya merupakan sebagian kecil dari kegiatan menganalisa berkas log. Untuk sistem yang cukup ramai, misalnya sebuah perguruan tinggi dengan jumlah pemakai yang ribuan, analisa berkas log merupakan satu pekerjaan tersendiri (yang melelahkan). Untuk itu adanya tools yang dapat membantu administrator untuk memproses dan menganalisa berkas log merupakan sesuatu yang sangat penting. Ada beberapa tools sederhana yang menganalia berkas log untuk mengamati kegagalan (*invalid password*, *login failure*, dan sebagainya) kemudian memberikan ringkasan. Tools ini dapat dijalankan setiap pagi dan mengirimkan hasilnya kepada administrator.

Backup secara rutin

Seringkali tamu tak diundang (*intruder*) masuk ke dalam sistem dan merusak sistem dengan menghapus berkas-berkas yang dapat ditemui. Jika intruder ini berhasil menjebol sistem dan masuk sebagai super user (administrator), maka ada kemungkinan dia dapat menghapus seluruh berkas. Untuk itu, adanya backup yang dilakukan secara rutin merupakan sebuah hal yang esensial. Bayangkan apabila yang dihapus oleh tamu ini adalah berkas penelitian, tugas akhir, skripsi, yang telah dikerjakan bertahun-tahun.

Untuk sistem yang sangat esensial, secara berkala perlu dibuat backup yang letaknya berjauhan secara fisik. Hal ini dilakukan untuk menghindari hilangnya data akibat bencana seperti kebakaran, banjir, dan lain sebagainya. Apabila data-data dibackup akan tetapi diletakkan pada lokasi yang sama, kemungkinan data akan hilang jika tempat yang bersangkutan mengalami bencana seperti kebakaran.

Penggunaan Enkripsi untuk meningkatkan keamanan

Salah satu mekanisme untuk meningkatkan keamanan adalah dengan menggunakan teknologi enkripsi. Data-data yang anda kirimkan diubah sedemikian rupa sehingga tidak mudah disadap. Banyak servis di Internet yang masih menggunakan "*plain text*" untuk *authentication*, seperti penggunaan pasangan userid dan password. Informasi ini dapat dilihat dengan mudah oleh program penyadap (*sniffer*).

Contoh servis yang menggunakan plain text antara lain:

- akses jarak jauh dengan menggunakan telnet dan rlogin
- transfer file dengan menggunakan FTP
- akses email melalui POP3 dan IMAP4
- pengiriman email melalui SMTP
- akses web melalui HTTP

Penggunaan enkripsi untuk remote akses (misalnya melalui ssh sebagai penggani telnet atau rlogin) akan dibahas di bagian tersendiri.

Telnet atau shell aman

Telnet atau *remote login* digunakan untuk mengakses sebuah "*remote site*" atau komputer melalui sebuah jaringan komputer. Akses ini dilakukan dengan menggunakan hubungan TCP/IP dengan menggunakan userid dan password. Informasi tentang userid dan password ini dikirimkan melalui jaringan komputer secara terbuka. Akibatnya ada kemungkinan seorang yang nakal melakukan "sniffing" dan mengumpulkan informasi tentang pasangan userid dan password ini[1].

1. Meskipun cara ini biasanya membutuhkan akses "root".

Untuk menghindari hal ini, enkripsi dapat digunakan untuk melindungi adanya sniffing. Paket yang dikirimkan dienkripsi dengan RSA atau IDEA sehingga tidak dapat dibaca oleh orang yang tidak berhak. Salah satu implementasi mekanisme ini adalah SSH (Secure Shell). Ada beberapa implementasi SSH ini, antara lain:

- ssh untuk UNIX (dalam bentuk source code, gratis)
- SSH untuk Windows95 dari Data Fellows (komersial)
  http://www.datafellows.com/
- TTSSH, yaitu skrip yang dibuat untuk *Tera Term Pro* (gratis, untuk Windows 95)
  http://www.paume.itb.ac.id/rahard/koleksi
- SecureCRT untuk Windows95 (shareware / komersial)

BAB 5

Keamanan Sistem World Wide Web

World Wide Web (WWW atau Web) merupakan salah satu "killer applications" yang menyebabkan populernya Internet. WWW dikembangkan oleh Tim Berners-Lee ketika sabatical di CERN.

Kehebatan WWW adalah kemudahan untuk mengakses informasi, yang dihubungkan satu dengan lainnya melalui konsep hypertext. Informasi dapat tersebar di mana-mana di dunia dan terhubung melalui hyperlink. Informasi lebih lengkap tentang WWW dapat diperoleh di <http://www.w3.org>.

Berkembangnya WWW dan Internet menyebabkan pergerakan sistem informasi untuk menggunakannya sebagai basis. Untuk itu, keamanan sistem informasi yang berbasis WWW dan teknologi Internet bergantung kepada keamanan sistem WWW tersebut.

Sistem WWW terdiri dari dua sisi: server dan client. Sistem server dan client memiliki permasalahan yang berbeda. Keduanya akan dibahas secara terpisah.

Keamanan Server WWW

Keamanan server WWW biasanya merupakan masalah dari seorang administrator. Dengan memasang server WWW di sistem anda, maka anda membuka akses (meskipun secara terbatas) kepada orang luar. Apabila server anda terhubung ke Internet dan memang server WWW anda disiapkan untuk publik, maka anda harus lebih berhati-hati.

Server WWW menyediakan fasilitas agar client dari tempat lain dapat mengambil informasi dalam bentuk berkas (file), atau mengeksekusi perintah (menjalankan program) di server. Fasilitas pengambilan berkas dilakukan dengan perintah "GET", sementara mekanisme untuk mengeksekusi perintah di server dikenal dengan istilah "CGI-bin" (Common Gateway Interface). Kedua servis di atas memiliki potensi lubang keamanan yang berbeda.

Adanya lubang keamanan di sistem WWW dapat dieksploitasi dalam bentuk yang beragam, antara lain:

- informasi yang ditampilkan di server diubah sehingga dapat mempermalukan perusahaan atau organisasi anda;
- informasi yang semestinya dikonsumsi untuk kalangan terbatas (misalnya laporan keuangan, strategi perusahaan anda, atau database client anda) ternyata berhasil disadap oleh saingan anda;
- informasi dapat disadap (seperti misalnya pengiriman nomor kartu kredit untuk membeli melalui WWW);
- server anda diserang sehingga tidak bisa memberikan layanan ketika dibutuhkan (denial of service attack);
- untuk server web yang berada di belakang firewall, lubang keamanan di server web yang dieksploitasi dapat melemahkan atau bahkan menghilangkan fungsi dari firewall.

Sebagai contoh serangan dengan mengubah isi halaman web, beberapa server Web milik pemerintah Indonesia sempat menjadi target serangan dari beberapa pengacau (dari Portugal) yang tidak

suka dengan kebijaksanaan pemerintah Indonesia dalam masalah Timor Timur. Mereka mengganti halaman muka dari beberapa server Web milik pemerintah Indonesia dengan tulisan-tulisan anti pemerintah Indonesia. Selain itu, beberapa server yang dapat mereka serang diporakporandakan dan dihapus isi disknya. Beberapa server yang sempat dijebol antara lain: server Departemen Luar Negeri, Hankam, Ipteknet, dan BPPT.

Kontrol Akses

Sebagai penyedia informasi (dalam bentuk berkas-berkas), sering diinginkan pembatasan akses. Misalnya, diinginkan agar hanya orang-orang tertentu yang dapat mengakses berkas (informasi) tertentu. Pada prinsipnya ini adalah masalah kontrol akses. Pembatasan akses dapat dilakukan dengan:

- membatasi domain atau nomor IP yang dapat mengakses;
- menggunakan pasangan userid & password;
- mengenkripsi data sehingga hanya dapat dibuka (dekripsi) oleh orang yang memiliki kunci pembuka.

Mekanisme untuk kontrol akses ini bergantung kepada program yang digunakan sebagai server. Salah satu caranya akan diuraikan pada bagian berikut.

Proteksi halaman dengan menggunakan password

Salah satu mekanisme mengatur akses adalah dengan menggunakan pasangan *userid* (*user identification*) dan *password*. Untuk server Web yang berbasis Apache[1], akses ke sebuah halaman (atau sekumpulan berkas yang terletak di sebuah directory di sistem Unix) dapat diatur dengan menggunakan berkas "`.htaccess`". Sebagai contoh, isi dari berkas tersebut dapat berupa:

```
AuthUserFile /home/budi/.passme
AuthGroupFile /dev/null
```

1. Mekanisme ini juga berlaku di server yang menggunakan program NCSA httpd dan CERN httpd.

```
AuthName “Khusus untuk Tamu Budi”
AuthType Basic
<Limit GET>
       require user tamu
</Limit>
```

Dalam contoh di atas, untuk mengakses direktori tersebut dibutuhkan userid “tamu” dan password yang sama dengan entry userid budi di berkas “/home/budi/.passme”. Ketika direktori tersebut diakses, akan muncul sebuah pop-up window yang menanyakan userid dan password.

Password di dalam berkas “/home/budi/.passme” dapat dibuat dengan menggunakan program “htpasswd”.

```
unix% htpasswd -c /home/budi/.passme budi
New password: *****
```

Secure Socket Layer

Salah satu cara untuk meningkatkan keamanan server WWW adalah dengan menggunakan enkripsi pada komunikasi pada tingkat socket. Dengan menggunakan enkripsi, orang tidak bisa menyadap data-data yang dikirimkan dari/ke server WWW. Salah satu mekanisme yang cukup populer adalah dengan menggunakan *Secure Socket Layer* (SSL) yang mulanya dikembangkan oleh *Netscape*.

Selain server WWW dari Netscape, beberapa server lain juga memiliki fasilitas SSL juga. Server WWW *Apache* (yang tersedia secara gratis) dapat dikonfigurasi agar memiliki fasilitas SSL dengan menambahkan software tambahan (SSLeay - yaitu implementasi SSL dari Eric Young - atau OpenSSL - yaitu implementasi Open Source dari SSL). Bahkan ada sebuah perusahaan (*Stronghold*) yang menjual Apache dengan SSL.

Penggunaan SSL memiliki permasalahan yang bergantung kepada lokasi dan hukum yang berlaku. Hal ini disebabkan:

- Pemerintah melarang ekspor teknologi enkripsi (kriptografi).

- Paten *Public Key Partners* atas *Rivest-Shamir-Adleman* (*RSA*) public-key cryptography yang digunakan pada SSL.

Oleh karena hal di atas, implementasi SSLeay Eric Young tidak dapat digunakan di Amerika Utara (Amerika dan Kanada) karena "melanggar" paten RSA dan RC4 yang digunakan dalam implementasinya. SSLeay dapat diperoleh dari:

- `http://www.psy.uq.oz.au/~ftp/Crypto`

Informasi lebih lanjut tentang SSL dapat diperoleh dari:

- `http://home.netscape.com/newsref/std`

Mengetahui Jenis Server

Informasi tentang server yang digunakan dapat digunakan oleh perusak untuk melancarkan serangan sesuai dengan tipe server dan operating system yang digunakan.

Informasi tentang program server yang digunakan sangat mudah diperoleh. Program *Ogre* (yang berjalan di sistem Windows) dapat mengetahui program server web yang digunakan. Sementara itu, untuk sistem UNIX, program *lynx* dapat digunakan untuk melihat jenis server dengan menekan kunci "sama dengan" (=).

Keamanan Program CGI

Meskipun mekanisme CGI tidak memiliki lubang keamanan, program atau skrip yang dibuat sebagai CGI dapat memiliki lubang keamanan. Misalnya, seorang pemakai yang nakal dapat memasang skrip CGI sehingga dapat mengirimkan berkas password kepada pengunjung yang mengeksekusi CGI tersebut.

Keamanan client WWW

Dalam bagian terdahulu dibahas masalah yang berhubungan dengan server WWW. Dalam bagian ini akan dibahas masalah-masalah yang berhubungan dengan keamanan client WWW, yaitu pemakai (pengunjung) biasa.

Seorang pengunjung sebuah tempat WWW dapat diganggu dengan berbagai cara. Misalnya, tanpa dia ketahui, dapat saja program Java, Javascript, atau ActiveX yang jahat didownload dan dijalankan. Program ini dapat mengirimkan informasi tentang anda ke server WWW tersebut, atau bahkan menjalankan program tertentu di komputer anda. Bayangkan apabila yang anda download adalah virus atau trojan horse yang dapat menghapus isi harddisk anda.

Bahan Bacaan

Informasi lebih lanjut mengenai keamanan sistem WWW dapat diperoleh dari sumber on-line sebagai berikut.

- <http://www.w3.org/Security/Faq/>

BAB 6

Eksploitasi Keamananan

Dalam bab ini akan dibahas beberapa contoh eksploitasi lubang keamanan. Contoh-contoh yang dibahas ada yan bersifat umum dan ada yang bersifat khusus untuk satu jenis operating system tertentu, atau untuk program tertentu dengan versi tertentu. Biasanya lubang keamanan ini sudah ditutup pada versi baru dari paket program tersebut sehingga mungkin tidak dapat anda coba.

Denial of Service Attack

"*Denial of Service (DoS) attack*" merupakan sebuah usaha (dalam bentuk serangan) untuk melumpuhkan sistem yang dijadikan target sehingga sistem tersebut tidak dapat menyediakan servis-servisnya (*denial of service*) atau tingkat servis menurun dengan drastis. Cara untuk melumpuhkan dapat bermacam-macam dan akibatnyapun dapat beragam. Sistem yang diserang dapat menjadi "bengong" (*hang*, *crash*), tidak berfungsi, atau turun kinerjanya (beban CPU tinggi).

Serangan denial of service berbeda dengan kejahatan pencurian data atau kejahatan memonitor informasi yang lalu lalang. Dalam serangan DoS tidak ada yang dicuri. Akan tetapi, serangan DoS dapat mengakibatkan kerugian finansial. Sebagai contoh apabila sistem yang diserang merupakan server yang menangani transaksi "*commerce*", maka apabila server tersebut tidak berfungsi, transaksi tidak dapat dilangsungkan. Bayangkan apabila sebuah bank diserang oleh bank saingan dengan melumpuhkan outlet ATM (Anjungan Tunai Mandiri, *Automatic Teller Machine*) yang dimiliki oleh bank tersebut. Atau sebuah credit card merchant server yang diserang sehingga tidak dapat menerima pembayaran melalui credit card.

Selain itu, serangan DoS sering digunakan sebagai bagian dari serangan lainnya. Misalnya, dalam serangan *IPspoofing* (seolah serangan datang dari tempat lain dengan nomor IP milik orang lain), seringkali DoS digunakan untuk membungkam server yang akan di*spoof*.

Land attack

Land attack merupakan serangan kepada sistem dengan menggunakan program yang bernama "*land*". Apabila serangan diarahkan kepada sistem Windows 95, maka sistem yang tidak diproteksi akan menjadi *hang* (dan bisa keluar layar biru). Demikian pula apabila serangan diarahkan ke beberapa jenis UNIX versi lama, maka sistem akan *hang*. Jika serangan diarahkan ke sistem Windows NT, maka sistem akan sibuk dengan penggunaan CPU mencapai 100% untuk beberapa saat sehingga sistem terlihat seperti macet. Dapat dibayangkan apabila hal ini dilakukan secara berulang-ulang. Serangan land ini membutuhkan nomor IP dan nomor port dari server yang dituju. Untuk sistem Windows, biasanya port 139 yang digunakan untuk menyerang.

Program land menyerang server yang dituju dengan mengirimkan packet palsu yang seolah-olah berasal dari server yang dituju. Dengan kata lain, source dan destination dari packet dibuat seakan-

akan berasal dari server yang dituju. Akibatnya server yang diserang menjadi bingung.

```
unix# ./land 192.168.1.1 139
land.c by m3lt, FLC
192.168.1.1:139 landed
```

Latierra

Program *latierra* merupakan "perbaikan" dari program land, dimana port yang digunakan berubah-ubah sehingga menyulitkan bagi pengamanan.

```
latierra v1.0b by MondoMan (elmondo@usa.net), KeG
Enhanced version of land.c originally developed by m3lt, FLC
Arguments:
* -i dest_ip = destination ip address such as 1.1.1.1
       If last octet is '-', then the address will increment
       from 1 to 254 (Class C) on the next loop
       and loop must be > 1 or -5 (forever).
       Alternatives = zone=filename.txt or list=filename.txt
       (ASCII) For list of alternative options,
       use -a instead of -h.
* -b port# = beginning port number (required).
-e port# = ending port number (optional)
-t = tcp flag options (f=fin, ~s=syn, r=reset, ~p=push, a=ack,
       u=urgent)
-v = time_to_live value, default=255
-p protocol = ~6=tcp, 17=udp, use -p option for complete list
-w window_size = value from 0 to ?, default=65000
-q tcp_sequence_number, default=3868
-m message_type
         (~0=none, 1=Out-Of-Band, 4=Msg_DontRoute
-s seconds = delay between port numbers, default=1
-o 1 = supress additional output to screen, default=0
-l loop = times to loop through ports/scan, default=1,
       -5=forever
* = required     ~ = default parameter values


unix# ./latierra -i 192.167.1.1 -b 139 -e 141

latierra v1.0b by MondoMan (elmondo@usa.net), KeG
Enhanced version of land.c originally developed by m3lt, FLC
```

```
Settings:
  (-i)   Dest. IP Addr   : 192.168.1.1
  (-b)   Beginning Port #: 139
  (-e)   Ending Port #   : 141
  (-s)   Seconds to Pause: 1
  (-l)   Loop            : 1
  (-w)   Window size     : 65000
  (-q)   Sequence Number : F1C (3868)
  (-v)   Time-to-Live    : 255
  (-p)   IP Protocol #   : 6
  (-t)   TCP flags       : syn push
Done.
```

Ping-o-death

Ping-o-death sebetulnya adalah eksploitasi program *ping* dengan memberikan packet yang ukurannya besar ke sistem yang dituju. Beberapa sistem UNIX ternyata menjadi hang ketika diserang dengan cara ini. Program ping umum terdapat di berbagai operating system, meskipun umumnya program ping tersebut mengirimkan packet dengan ukuran kecil (tertentu) dan tidak memiliki fasilitas untuk mengubah besarnya packet. Salah satu implementasi program ping yang dapat digunakan untuk mengubah ukuran packet adalah program ping yang ada di sistem Windows 95.

Ping broadcast (smurf)

Salah satu mekanisme serangan yang baru-baru ini mulai marak digunakan adalah menggunakan ping ke alamat *broadcast*, ini yang sering disebut dengan *smurf*. Seluruh komputer (*device*) yang berada di alamat broadcast tersebut akan menjawab. Apakah ini merupakan standar?

Jika sebuah sistem memiliki banyak komputer (*device*) dan ping broadcast ini dilakukan terus menerus, jaringan dapat dipenuhi oleh respon-respon dari device-device tersebut. Akibatnya jaringan menjadi lambat.

```
$ ping 192.168.1.255
PING 192.168.1.255 (192.168.1.255): 56 data bytes
64 bytes from 192.168.1.4: icmp_seq=0 ttl=64 time=2.6 ms
64 bytes from 192.168.1.2: icmp_seq=0 ttl=255 time=24.0 ms
(DUP!)
64 bytes from 192.168.1.4: icmp_seq=1 ttl=64 time=2.5 ms
64 bytes from 192.168.1.2: icmp_seq=1 ttl=255 time=4.7 ms
(DUP!)
64 bytes from 192.168.1.4: icmp_seq=2 ttl=64 time=2.5 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=255 time=4.7 ms
(DUP!)
64 bytes from 192.168.1.4: icmp_seq=3 ttl=64 time=2.5 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=255 time=4.7 ms
(DUP!)
--- 192.168.1.255 ping statistics ---
4 packets transmitted, 4 packets received, +4 duplicates, 0%
packet loss
round-trip min/avg/max = 2.5/6.0/24.0 ms
```

Smurf attack biasanya dilakukan dengan menggunakan *IP spoofing*, yaitu mengubah nomor IP dari datangnya *request*, tidak seperti contoh di atas. Dengan menggunakan *IP spoofing*, respon dari *ping* tadi dialamatkan ke komputer yang IPnya di*spoof*. Akibatnya komputer tersebut akan menerima banyak paket. Hal ini dapat mengakibatkan pemborosan penggunaan (bandwidth) jaringan yang menghubungkan komputer tersebut. Dapat dibayangkan apabila komputer yang dispoof tersebut memiliki hubungan yang berkecepatan rendah dan ping diarahkan ke sistem yang memiliki banyak host. Hal ini dapat mengakibatkan DoS attack.

Sniffer

Program sniffer adalah program yang dapat digunakan untuk menyadap data dan informasi melalui jaringan komputer.

Sniffit

Program sniffit dijalankan dengan userid root (atau program dapat di-setuid root sehingga dapat dijalankan oleh siapa saja) dan dapat menyadap data. Untuk contoh penggunaan sniffit, silahkan baca dokumentasi yang menyertainya. (Versi berikut dari buku ini akan menyediakan informasi tentang penggunaannya.)

BAB 7

Cyberlaw: Hukum dan Keamanan

A man has a right to pass through this world, if he wills,
without having his picture published, his business enterprise discussed,
his successful experiments written up for the benefit of others,
or his eccentricities commented upon,
whether in handbills, circulars, catalogues, newspapers or periodicals.
-- Chief Justice Alton B. Parker (New York Court of Appeals),
decision in Roberson v. Rochater Folding Box Co., 1901

Masalah keamanan erat hubungannya dengan masalah hukum. Dalam bab ini akan diulas beberapa aspek keamanan yang berhubungan dengan masalah hukum.

Internet menghilangkan batas tempat dan waktu, dua asas yang cukup esensial di bidang hukum. Terhubungnya sebuah sistem informasi dengan Internet membuka peluang adanya kejahatan melalui jaringan komputer. Hal ini menimbulkan tantangan bagi penegak hukum. Hukum dari sebagian besar negara di dunia belum menjangkau daerah cyberspace. Saat ini hampir semua negara di dunia berlomba-lomba untuk menyiapkan landasan hukum bagi Internet. Tentunya banyak hal yang dapat dibahas, akan tetapi dalam buku ini hanya dibahas hal-hal yang berkaitan dengan

masalah keamanan (*security*), masalah lain seperti pajak (hal-hal yang berhubungan dengan perbankan dan bisnis) yang tidak langsung terkait dengan masalah keamanan tidak dibahas di dalam buku ini.

Dalam aplikasi e-commerce, misalnya, ada masalah yang berkaitan dengan hukum yaitu masalah privacy dan penggunaan teknologi kriptografi (seperti penggunaan enkripsi). Setiap negara memiliki hukum yang berlainan. Misalnya negara Amerika Serikat melarang ekspor teknologi enkripsi. Selain itu sistem perbankan setiap negara memiliki hukum yang berlainan.

Penegakan hukum (*law enforcement*) merupakan masalah tersendiri. Ambil contoh seseorang yang tertangkap basah melakukan cracking yang mengakibatkan kerugian finansial. Hukuman apa yang dapat diberikan? Sebagai contoh, di Cina terjadi hukuman mati atas dua orang crackers yang tertangkap mencuri uang sebesar US$31.400 dari sebuah bank di Cina bagian Timur. Berita lengkapnya dapat dibaca di:

- http://www.news.com/News/Item/0,4,30332,00.html
- http://cnn.com/WORLD/asiapcf/9812/28/BC-CHINA-HACKERS.reut/index.html
- http://slashdot.org/articles/98/12/28/096231.shtml

Penggunaan Enkripsi dan Teknologi Kriptografi Secara Umum

Salah satu cara untuk mengamankan data dan informasi adalah dengan menggunakan teknologi kriptografi (*cryptography*). Misalnya data dapat dienkripsi dengan menggunakan metoda tertentu sehingga hanya dapat dibaca oleh orang tertentu. Ada beberapa masalah dalam penggunaan teknologi kriptografi ini, antara lain:

- Dilarangnya ekspor teknologi kriptografi dari Amerika Serikat (USA), padahal teknologi yang canggih ini banyak dikembangkan di USA. Adanya larangan ini membuat *interoperability* antar produk yang menggunakan teknologi kriptografi menjadi lebih sulit. Hal yang lain adalah selain negara Amerika, negara lain mendapat produk dengan kualitas keamanan yang lebih rendah. Sebagai contoh, Web browser Netscape dilengkapi dengan fasilitas security dengan menggunakan sistem RSA. Pada saat buku ini ditulis, implementasi RSA dengan menggunakan 128 bit hanya dapat digunakan di dalam negeri Amerika saja (tidak boleh diekspor). Untuk itu Netscape harus membuat versi Internasional yang hanya menggunakan 56 bit dan boleh diekspor. Tingkat keamanan sistem yang menggunakan 56 bit lebih rendah dibandingkan dengan sistem yang menggunakan 128 bit.
- Bagi sebuah negara, ketergantungan masalah keamanan kepada negara lain merupakan suatu aspek yang cukup sensitif. Kemampuan negara dalam mengusai teknologi merupakan suatu hal yang esensial. Ketergantungan kepada negara lain ini juga sangat penting dilihat dari sudut bisnis karena misalnya jika *electronic commerce* menggunakan produk yang harus dilisensi dari negara lain maka banyak devisa negara yang akan tersedot hanya untuk melisensi teknologi tersebut.
- Algoritma-algoritma yang sangat baik untuk kriptografi umumnya dipatenkan. Hal ini seringkali mempersulit implementasi sebuah produk tanpa melanggar hak patent. Selain itu setiap negara di dunia memiliki pandangan tertentu terhadap hak patent. Sebagi contoh, algoritma RSA dipatenkan di Amerika Serikat akan tetapi tidak diakui di Jepang (lihat cerita latar belakangnya di [11]).

Pemerintah negara tertentu berusaha untuk menggunakan peraturan (regulation) untuk mengatur penggunaan teknologi enkripsi. Hal ini ditentang dan diragukan oleh banyak pihak. Dalam sebuah survey [15], 82% responden menyatakan bahwa pemerintah tidak dapat mengatur secara efektif penyebaran penggunaan teknologi enkripsi melalui regulasi.

Masalah yang berhubungan dengan patent

Enkripsi dengan menggunakan public key sangat membantu dalam meningkatkan keamanan informasi. Salah satu algoritma yang cukup populer digunakan adalah RSA. Algoritma ini dipatenkan di Amerika Serikat dengan nomor U.S. Patent 4,405,829 yang dikeluarkan pada tanggal 20 Agustus 1983. Paten yang dimiliki oleh *Public Key Partners* (PKP, Sunnyvale, California) ini akan habis di tahun 2000. RSA tidak dipatenkan di luar daerah Amerika Utara. Bagaimana dengan penggunaan algoritma RSA ini di Indonesia? Penggunaan enkripsi di luar Amerika ini merupakan sebuah topik diskusi yang cukup seru.

Privacy

Aspek privacy sering menjadi masalah yang berkaitan dengan masalah keamanan. Pemakai (*user*) umumnya ingin informasi dan kegiatan yang dilakukannya tidak diketahui oleh orang lain, termasuk oleh administrator. Sementara itu, demi menjaga keamanan dan tingkat performance dari sistem yang dikelolanya, seorang administrator seringkali harus mengetahui apa yang dilakukan oleh pemakai sistemnya.

Sebagai contoh kasus, seorang administrator merasa bahwa salah satu pemakainya mendapat serangan mailbomb dari orang lain dengan mengamati jumlah dan ukuran email yang diterima sang pemakai. Adanya serangan mailbomb ini dapat menurunkan performance sistem yang dikelolanya, bahkan bisa jadi server yang digunakan bisa menjadi macet (*hang*). Kalau server macet, berarti pemakai lain tidak dapat mengakses emailnya. Masalahnya, untuk memastikan bahwa pemakai yang bersangkutan mengalami serangan mailbomb administrator harus melihat (mengintip?) email dari sang pemakai tersebut. Hal ini menjadi pertanyaan, karena hal ini dapat dianggap melanggar privacy dari pemakai yang bersangkutan.

Salah satu topik yang sering berhubungan dengan privacy adalah penggunaakn "*key escrow*" atau "*key-recovery system*", dimana pemerintah dapat membuka data yang sudah terenkripsi dengan kunci khusus. Masyarakat umumnya tidak setuju dengan penggunaan key-recovery system ini, seperti diungkapkan dalam survey IEEE Computer [15]: "*77% of members agree that key-recovery systems make it too easy for government to access encrypted data without permission*."

BAB 8

Referensi

Daftar Bahan Bacaan

1. Steven M. Bellovin, "*Security Problems in TCP/IP Protocol Suite,*" Computer Communication Review, Vol. 19, No. 2, pp. 32-48, 1989.
2. Lawrie Brown, "*Lecture Notes for Use with Network and Internetwork Security by William Stallings,*" on-line document.
   <http://www1.shore.net/~ws/Security-Notes/index.html>
3. CERT, "*CERT Advisory, CA-99-01-Trojan-TCP-Wrappers,*" 21 Januari 1999.
   <http://www.cert.org/advisories/CA-99-01-Trojan-TCP-Wrappers.html>
4. Bill Cheswick, "*An Evening with Berferd: in which a cracker is lured, endured, and studied,*" 1991.
5. Computer Security Institute, "*1999 CSI/FBI Computer Crime and Security Survey,*" CSI, Winter 1999.
   <http://www.gocsi.com>
6. Patrick W. Dowd, and John T. McHenry, "Network Security: It's Time To Take It Seriously," *IEEE Computer*, pp. 24-28, September 1998.

7. Electronic Frontier Foundation, "*Cracking DES: Secrets of Encryption Research, Wiretap Politics and Chip Design*," O'Reilly & Associates, 1998. ISBN 1-56592-520-3.
8. Sidnie Feit, "*SNMP: A guide to network management*," McGraw-Hill, 1995.
9. Warwick Ford, and Michael Baum, "*Secure Electronic Commerce: building infrastructure for digital signatures & encryption*," Prentice Hall PTR, 1997.
10. Fyodor, "*Remote OS detection via TCP/IP Stack FingerPrinting*," 18 Oktober 1998. Merupakan bagian dari paket program Nmap.
11. Simson Garfinkel, "*PGP: Pretty Good Privacy*," O'Reilly & Associates, Inc., 1995.
12. Simson Garfinkel, and Gene Spafford, "*Practical UNIX & Internet Security*," O'Reilly & Associates, Inc., 2nd edition, 1996.
13. John D. Howard, "*An Analysis Of Security Incidents On The Internet 1989 - 1995*," PhD thesis, Engineering and Public Policy, Carnegie Mellon University, 1997.
14. David J. Icove, "Collaring the cybercrook: an investigator's view," *IEEE Spectrum*, pp. 31-36, June 1997.
15. IEEE Computer, "Members React to Privacy dan Encryption Survey," *IEEE Computer*, pp. 12-15, September 1998.
16. Anna Johnson, "Companies Losing Millions over Rising Computer Crime," *Shake Security Journal*, March, 1998. http://www.shake.net/crime_march98.htm
17. J. Kriswanto, "*Bidang Jaringan dan Electronic Commerce Nusantara-21*," Yayasan Litbang Telekomunikasi Informatika (YLTI), Departemen Pariwisata, Pos dan Telekomunikasi, Maret, 1998.
18. Jonathan Littman, "*The Fugitive Game: online with Kevin Mitnick*," Little Brown, 1996.
19. Cricket Liu, Jerry Peek, Russ Jones, Bryan Buus, and Adrian Nye, "*Managing Internet Information Services*," O'Reilly & Associates, Inc., 1994.
20. Richard Morin, "DES Verites," *SunExpert Magazine*, pp. 32-35, October 1998.

21. Budi Rahardjo, "*Keamanan Sistem Informasi: Beberapa Topik Keamanan di Internet*," Seminar Informasi Infrastruktur Nasional, ITB, 1997.
22. Budi Rahardjo, "Keamanan Sistem Internet," *Pikiran Rakyat*, 3 Maret 1998.
23. Budi Rahardjo, "Mengimplementasikan Electronic Commerce di Indonesia," *Technical Repor*t, PPAU Mikroelektronika ITB, 1999.
24. Marcus Ranum "*Thinking About Firewalls.*" ftp://ftp.tis.com/pub/firewalls/firewall.ps.Z
25. RFC2196 - Site Security Handbook (B. Fraser, editor)
26. Bruce Schneier, "*Applied Cryptography: Protocols, Algorithms, and Source Code in C,*" second edition, John Wiley & Sons, Inc., 1996.
27. William Stallings, "*Network and Internetwork Security*," Prentice Hall, 1995.
28. Paul Taylor, "*Them and us*", electronic document (Chapter 6 of his PhD dissertation), 1997. http://www.rootshell.com
29. Tim Koordinasi Telematika Indonesia, "*Gambaran Umum Pembangunan Telematika Indonesia,*" 1998.

Sumber informasi dan organisasi yang berhubungan dengan keamanan sistem informasi

1. *2600*
   http://www.2600.com
   Berisi informasi tentang bermacam-macam hacking bawah tanah beserta koleksi gambar dari tempat-tempat (web site) yang pernah dihack.
2. *Anti Online*
   http://www.antionline.com

3. *CERT (Center of Emergency Response Team)*
http://www.cert.org
Merupakan sumber informasi yang cukup akurat dan up to date tentang keamanan Internet. CERT Advisories merupakan pengumuman berkala tentang security hole and cara mengatasinya.

4. CIAC
ftp://ciac.llnl.gov/pub/ciac

5. *COAST (Computer Operations, Audit, and Security Technology)*
http://www.cs.purdue.edu/coast/coast.html
Berisi informasi tentang riset, tools, dan informasi yang berhubungan dengan masalah keamanan.

6. *CSI (Computer Security Institute)*
http://www.gocsi.com
Hasil survey, materi seminar.

7. *Electronic Privacy Information Center*
http://www.epic.org

8. *ICSA (International Computer Security Association)*
http://www.icsa.net/

9. *ID-CERT (Indonesia CERT)*
http://www.paume.itb.ac.id/rahard/id-cert
http://id-cert.internet.co.id
http://idcert.regex.com (akan datang)
Seperti CERT akan tetapi dikhususkan untuk domain Indonesia.

10. *NEC*
ftp://ftp.nec.com/pub/security

11. *OpenSec.Net*
http://www.opensec.net
Berisi koleksi software tools yang berhubungan dengan masalah keamaman. Saat ini lebih uptodate daripada Rootshell.

12. *Packet Storm*
http://www.genocide2600.com/~tattooman/new.shtml
Berisi koleksi software yang berhubungan dengan security.

13. *RISK: Electronic Digest*
http://catless.ncl.ac.uk/Risks

14. *Rootshell*
    http://www.rootshell.com
    http://rootshell.connectnet.com/docs/
    Berisi informasi terbaru tentang lubang keamanan, program-program yang dapat digunakan untuk menguji atau eksploitasi keamanan, dan juga menyimpan tulisan (makalah, tutorial, artikel, dsb.) tentang sistem keamanan. Note: saat ini web ini sudah jarang diupdate.
15. SANS
    http://www.sans.org
16. Security Portal
    http://www.securityportal.com/
    Berisi artikel dan berita yang berhubungan dengan keamaman.
17. TAMU
    ftp://net.tamu.edu/pub/security

Daftar perusahaan yang berhubungan dengan keamanan

1. *Data Fellows*
   http://www.datafellows.com/
   Menyediakan SSH (secure shell), server dan client, untuk sistem UNIX dan Windows. Juga menyediakan proteksi virus.
2. *PGP Internasional*
   http://www.pgpi.com
   Menyediakan implementasi PGP versi internasional (yang dapat digunakan di luar Amerika Serikat).
3. *Secure Networks*
   http://www.securenetworks.com

Sumber software / tools

1. Apache-SSL: versi web server Apache yang menggunakan SSL
   http://www.apache-ssl.org
2. Auditd: monitor and log system calls dari HERT
   ftp://ftp.hert.org/pub/linux/auditd
3. Autobuse: identifikasi abuse dengan memonitor logfile
   http://www.picante.com/~gtaylor/autobuse
4. Fwconfig: front end tool untuk ipfwadm
   http://www.mindstorm.com/~sparlin/fwconfig.shtml
5. GnuPG, GNU Privacy Guard
   http://www.d.shuttle.de/isil/gnupg
6. ipchains: Linux kernel packet filtering yang baru, yang akan menggantikan ipfwadm
   http:// www.rustcorp.com/linux/ipchains
7. ipfwadm: Linux kernel packet filtering yang lama
   http://www.xos.nl/linux/ipfwadm
8. IPlog: berisi iplog, icmplog, udplog
   http://www.ojnk.org/~eric
9. Karpski, network monitor berbasis GTK+
   http://mojo.calyx.net/~bxx/karpski.html
10. Ksniff
    http://www.mtco.com/~whoop/ksniff/ksniff.html
11. libpcap: library untuk menangkap (capture) packet
    ftp://ftp.ee.lbl.gov/libpcap.tar.Z
12. Nessus: security auditing tools (Linux)
    http://www.nessus.org
13. netwatch: monitor network, text-mode
14. nmap (UNIX): probing, OS fingerprinting
    http://www.insecure.org/nmap/
    http://www.dhp.com/~fyodor/nmap
15. ntop: memantau penggunaan jaringan
    http://jake.unipi.it/~deri/ntop/

16. OpenSec: koleksi tools
    http://www.opensec.net
17. OpenSSL: Open Source toolkit SSL v2/v3 dan Transport Layer Security (TLS v1)
    http://www.openssl.org
18. queso: OS fingerprinting
    http://www.apostols.org/projectz/
19. Retina: scanning Windows NT
    http://www.eeye.com
20. Saint
    http://www.wdsi.com/saint/
21. *SBScan*
    http://www.haqd.demon.co.uk/security.htm
22. Shadow: intrusion detection system dari SANS
    http://www.sans.org
23. SSLeay: free SSL crypto library & applications
    http://www.ssleay.org
24. snort (UNIX), packet logger
25. Socks, proxy server
    http://www.socks.nec.com
26. Squid: web proxy server
    http://squid. nlanr.net
27. TCP wrapper (UNIX), official site
    ftp://ftp.porcupine.org/pub/security/
28. tcplogd: memantau adanya probing
    http://www.kalug.lug.net
29. Trinux (Linux)
    http://www.trinux.org

Symbols

A

B

C

D

E

F

G

H

I

winuke 40
wu-ftpd 58