BUKU AJAR

TEKNIK ELEKTRO

SRI HARTANTO

KEAMANAN DAN KEHANDALAN JARINGAN

PENERBIT CV. MITRA ILMU

IKAPI
IKATAN PENERBIT INDONESIA
No. 041/SSL/I022

BUKU AJAR TEKNIK ELEKTRO

KEAMANAN DAN KEHANDALAN JARINGAN

BUKU AJAR TEKNIK ELEKTRO

KEAMANAN DAN KEHANDALAN JARINGAN

SRI HARTANTO

PENERBIT
CV MITRA ILMU
MAKASSAR
Anggota IKAPI No. 041/SSL/2022

Perpustakaan Nasional: Katalog Dalam Terbitan (KDT)

Sri Hartanto

Keamanan dan Kehandalan Jaringan/Sri Hartanto.
—Ed. 1, Cet. 1.—Makassar: Penerbit CV Mitra Ilmu, 2023.
viii, 140 hlm., 21 cm.
Bibliografi: hlm. 141
ISBN 978-623-145-089-0

2023.0519 SRI
Sri Hartanto

KEAMANAN DAN KEHANDALAN JARINGAN

Cetakan ke-1, Mei 2023

Editor : Ujang Wiharja
Setter : Sri Hartanto
Desain Cover : Sulaiman

Dicetak di Gilby Jaya Printing

PENERBIT CV MITRA ILMU

Anggota IKAPI

Kantor Pusat:
Jl. Kesatuan 3 No. 11 Kelurahan Maccini Parang
Kecamatan Makassar Kota Makassar 90144
Telpon : 081342345219
E-mail : mitrailmu@mitrailmumakasaar.com
http: //www. mitrailmumakasaar.com

KATA PENGANTAR

Alhamdulillahi robbil ‘alamiin.

Segala puji syukur dipanjatkan ke hadirat ALLAH SWT atas segala petunjuk, rahmat dan hidayah-Nya sehingga dapat disusun Buku Ajar Keamanan dan Kehandalan Jaringan ini sebagai bahan pembelajaran bagi mahasiswa yang mengikuti perkuliahan Keamanan dan Kehandalan Jaringan.

Tidak ada gading yang tidak retak, maka diharapkan saran dan masukan dari pembaca untuk perbaikan buku ini ke depannya. Diharapkan, buku ajar ini dapat bermanfaat bagi mahasiswa yang mempelajari dan mengikuti perkuliahan Keamanan dan Kehandalan Jaringan. Selain itu, buku ini juga ditujukan untuk memenuhi Tri Dharma Perguruan Tinggi dalam menyediakan Buku Ajar untuk Program Studi Teknik Elektro, khususnya untuk mata kuliah Keamanan dan Kehandalan Jaringan.

Terimakasih disampaikan kepada semua pihak yang telah membantu dalam penyusunan buku ini, baik pimpinan, karyawan maupun mahasiswa sehingga buku ini dapat diselesaikan dengan baik dan dapat diterbitkan.

Penyusun,

Sri Hartanto

DAFTAR ISI

BAB I
KONSEP DASAR

1.1. Pemahaman Keamanan dan Kehandalan Jaringan

Keamanan jaringan telekomunikasi adalah keamanan yang berkaitan dengan perlindungan jaringan telekomunikasi yang meliputi semua perangkat yang digunakan untuk melaksanakan komunikasi. Keamanan jaringan telekomunikasi bertujuan untuk mencegah kemungkinan terjadinya ancaman atau gangguan terhadap jalannya komunikasi (*preventive*). Selain itu, keamanan jaringan telekomunikasi juga meliputi pendeteksian dan perbaikan jaringan terhadap kemungkinan ancaman yang memasuki jaringan telekomunikasi (*detection and post attack recovery*).

Kehandalan jaringan telekomunikasi adalah bagian yang tidak terpisahkan dari keamanan jaringan telekomunikasi. Kehandalan jaringan telekomunikasi adalah kemampuan jaringan telekomunikasi untuk menyediakan layanan informasi dan komunikasi ke pengguna tanpa adanya gangguan baik dari sistem itu sendiri maupun gangguan karena faktor di luar sistem. Dengan terjaminnya kehandalan jaringan telekomunikasi, maka semua perangkat komunikasi yang digunakan dapat bekerja dengan baik dan memberikan tingkat kualitas layanan/*Quality of Service* (QoS) yang tinggi.

Keamanan jaringan telekomunikasi dapat diklasifikasikan menjadi:

1. Keamanan bersifat fisik (*physical security*), yang meliputi akses orang ke gedung, peralatan, dan media yang

digunakan.

2. Keamanan yang berhubungan dengan orang (*personnel*), yang meliputi identifikasi, dan profil resiko dari orang yang mempunyai akses (pekerja).
3. Keamanan yang berhubungan dengan informasi dan media serta teknik telekomunikasi, meliputi kelemahan pada perangkat lunak (*software*) yang digunakan untuk mengelola informasi.
4. Keamanan yang berhubungan dengan penggunaan atau operasionalisasi, meliputi kebijakan (*policy*) dan prosedur yang digunakan untuk mengatur dan mengelola sistem keamanan jaringan telekomunikasi, dan juga meliputi prosedur setelah ancaman keamanan (*post attack recovery*).

1.2. Aspek Keamanan Jaringan

Keamanan jaringan telekomunikasi yang berkaitan dengan sistem informasi meliputi aspek-aspek sebagai berikut:

1. *Privacy/Confidentiality*

   Merupakan aspek yang berkaitan dengan informasi seseorang dimana tidak semua orang berhak untuk mengaksesnya. *Privacy* berkaitan dengan informasi pribadi seperti nama, umur, alamat, nomor telepon dan sebagainya, sedangkan *confidentiality* berkaitan dengan informasi pribadi dalam bentuk suatu kepercayaan yang diberikan pihak lain, seperti data perbankan, data asuransi dan sebagainya.
2. *Integrity*

   Merupakan aspek yang menekankan keutuhan informasi dan menjamin informasi tidak diubah oleh orang yang

tidak berhak mengaksesnya. Ancaman keamanan jaringan telekomunikasi dapat berupa pihak lain yang mengubah informasi tanpa ijin, seperti misalnya suatu *email* yang dapat saja "ditangkap" (*intercept*) di tengah jalan, diubah isinya (*altered*, *tampered, modified*), kemudian diteruskan ke alamat yang tujuan. Untuk itu diperlukan adanya metode kriptografi yang menjamin integritas data pada saat pengiriman.

3. *Authentication*

   Merupakan aspek yang berhubungan dengan keaslian informasi, dimana informasi dijamin keaslian dan keabsahannya dan hanya orang yang berhak mengakses informasi yang dapat mengubah informasi. Dalam hal ini diperlukan adanya *personal identification number* (PIN) atau *password.*

4. *Availability*

   Aspek ini berhubungan dengan ketersediaan informasi ketika dibutuhkan. Sistem informasi yang diserang atau dijebol dapat menghambat atau meniadakan akses ke informasi. Ancaman keamanan dapat berupa: "*Denial of Service attack* (*DoS attack*), *mailbomb*.

5. *Access Control*

   Aspek ini berhubungan dengan cara pengaturan akses ke informasi. Hal ini biasanya berhubungan dengan klasifikasi informasi (*public, private, confidential, top secret*) dan pengguna (*guest, admin, top manager*, dan sebagainya).

6. *Non-repudiation*

   Aspek ini terdapat dalam perdagangan elektronik (*e-commerce*) yang menjamin seseorang tidak dapat

menyangkal telah melakukan suatu transaksi. Misalnya, seseorang yang mengirimkan *email* untuk memesan barang tidak dapat menyangkal bahwa dia telah mengirimkan *email* tersebut. Penggunaan *digital signature*, *certificates*, dan teknologi kriptografi secara umum dapat menjamin aspek ini.

1.3. Istilah Dalam Keamanan Jaringan

Istilah yang umum ditemui dalam keamanan jaringan adalah:

1. *Hacking* adalah setiap usaha atau kegiatan di luar ijin atau sepengetahuan pengelola jaringan telekomunikasi untuk memasuki suatu jaringan telekomunikasi dan mencoba mencuri *file* seperti *file password* dan sebagainya. Pelakunya disebut *hacker*, yaitu seorang atau sekumpulan orang yang berusaha menembus sistem pengamanan jaringan telekomunikasi.
2. *Cracking* adalah setiap usaha atau kegiatan untuk memasuki suatu jaringan telekomunikasi dan mencoba merusak dan menghancurkan integritas jaringan telekomunikasi tersebut. Pelakunya disebut *cracker*, yaitu seorang atau sekumpulan orang yang merusak dan menghancurkan integritas seluruh jaringan telekomunikasi.
3. *Denial of Service (*DoS*)* adalah usaha untuk membanjiri suatu *IP Address* dengan informasi sehingga menyebabkan *crash* atau jaringan telekomunikasi kehilangan hubungan ke internet. *Distributed Denial of Service (*DDoS*)* adalah serangan DoS yang menggunakan banyak perangkat komunikasi dalam jaringan telekomunikasi yang berbeda

untuk melancarkan ancaman keamanan jaringan telekomunikasi. Seorang *hacker* mengambilalih beberapa perangkat dan menggunakannya sebagai platform untuk menjalankan ancaman keamanan jaringan telekomunikasi.

4. *Theft of Information* adalah tindakan mencuri informasi rahasia dari suatu perusahaan dengan menggunakan program pembobol *password*, dan yang sejenisnya.
5. *Corruption of Data* adalah penyerangan dengan merusak data yang selama ini disimpan dalam *harddisk* suatu *host*.
6. *Spoofing* adalah suatu bentuk kegiatan pemalsuan dimana seorang *hacker* memalsukan (*to masquerade*) identitas seorang pengguna hingga dia berhasil *login* secara *illegal* ke dalam suatu jaringan telekomunikasi seolah-olah seperti pengguna yang asli.
7. *Sniffer* adalah kata lain dari "*network analyser*" yang berfungsi sebagai alat untuk memantau jaringan telekomunikasi. Alat ini dapat dioperasikan hampir pada seluruh jenis protokol seperti Ethernet, TCP/IP, IPX, dan lainnya.
8. *Password Cracker* adalah suatu program yang dapat membuka enkripsi suatu *password* atau sebaliknya untuk mematikan sistem pengamanan *password*.
9. *Destructive Devices* adalah sekumpulan program virus yang dibuat khusus untuk melakukan penghancuran informasi-informasi, di antaranya *Trojan*, *Worm, Email Bombs*, dan *Nukes*.
10. *Scanner* adalah suatu program yang secara otomatis akan mendeteksi kelemahan (*security weaknesses*) suatu perangkat dalam jaringan lokal (*local host*) ataupun perangkat dalam jaringan telekomunikasi di lokasi lain

(*remote host*). Oleh karena itu, dengan menggunakan program ini, seorang *hacker* yang secara fisik berada di Inggris dapat dengan mudah menemukan *security weaknesses* pada suatu *server* di Amerika ataupun di belahan dunia lainnya, tanpa harus meninggalkan ruangannya.

BAB II
KRIPTOGRAFI DAN STEGANOGRAFI

2.1. Metode Pengamanan Informasi

Untuk mengelola suatu jaringan telekomunikasi, perlu memperhatikan aspek pengamanan informasi (*information security*), dimana informasi hanya dapat diakses oleh pengelola (pembuat) informasi serta pihak-pihak yang diberi hak untuk mengaksesnya. Selain itu, perlu juga diperhatikan aspek perolehan informasi (*information intellegence*), sehingga hanya pihak-pihak yang memiliki hak akses informasi yang dapat mengetahui dan mencari keberadaan informasi tersebut. Metode pengamanan informasi dapat dibedakan atas Steganografi (*Steganography*) dan Kriptografi (*Cryptography*).

Steganografi (*Steganography*) adalah metode untuk membuat informasi sulit ditemukan, dengan menyembunyikan suatu informasi sehingga menjadi tidak terlihat (tidak dapat diakses). *Steganography* berasal dari bahasa Yunani, yaitu dari kata *stegano*, yang berarti tersembunyi dan *graphein*, yang berarti gambar yang mengandung pengertian tertentu (tulisan), sehingga steganografi didefinisikan secara lengkap sebagai metode untuk menyembunyikan tulisan (informasi). Dalam penerapannya, meskipun informasi berada pada media penyimpanan atau media transmisi yang dapat dikases, tetapi pihak lain tidak dapat menemukan dengan mudah informasi tersebut (*intelligence access*).

Kriptografi (*Cryptography*) adalah metode untuk membuat informasi yang dapat ditemukan (dapat diakses)

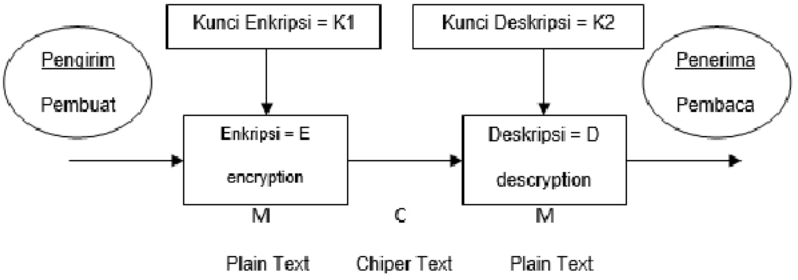
menjadi tidak dapat dikenali (dapat dibaca) karena informasi sudah diacak. *Cryptography* berasal dari bahasa Yunani, yaitu dari kata "*crypto*" berarti "*secret*" (rahasia) dan *graphein*, yang berarti gambar yang mengandung pengertian tertentu (tulisan). Dalam *cryptography*, pengamanan informasi dapat dilakukan dengan berbagai macam algoritma, dimana algoritma yang sederhana adalah dengan metode transposisi dan substitusi. Pada metode transposisi, susunan informasi diacak posisinya, sedangkan pada metode substitusi, informasi yang sebenarnya diganti dengan informasi lain. Untuk itu perlu digunakan kunci enkripsi untuk melakukan pengacakan posisi informasi atau penggantian suatu informasi dengan informasi lain, dan kunci dekripsi yang digunakan untuk menerjemahkan kembali informasi yang sudah diacak.

2.2. Pengenalan Kriptografi

Secara lebih rinci, kriptografi didefinisikan sebagai sekumpulan metode yang digunakan untuk mengubah informasi yang dapat dibaca (*plaintext*) menjadi informasi yang diacak (*ciphertext*) di sisi pengirim, dan di sisi penerima yang berhak mengakses, *ciphertext* dikembalikan menjadi *plaintext. Ciphertext* sebagai algoritma kriptografi merupakan persamaan matematika yang digunakan untuk melakukan proses enkripsi dan dekripsi.

Penggunaan kunci enkripsi dan dekripsi dapat dilihat dalam Gambar 2.1. berikut. Proses enkripsi (*encryption*) adalah proses mengubah *plaintext* menjadi *ciphertext*, sedangkan proses dekripsi (*decryption*) adalah proses mengembalikan *ciphertext* menjadi *plaintext*. Kunci enkripsi digunakan untuk menyandikan informasi sehingga tidak dapat diakses oleh

pihak yang tidak berhak. Untuk membuka kunci enkripsi digunakan kunci dekripsi yang algoritmanya dapat berbeda dengan kunci enkripsi sehingga kunci enkripsi dapat diketahui secara umum (*public key*). Selain itu, kunci dekripsi dapat juga memiliki algoritma yang sama seperti kunci enkripsi, sehingga membuat kunci enkripsi harus dijaga kerahasiaannya (*private key*).



Gambar 2.1. Penggunaan Kunci Enkripsi dan Dekripsi

Secara matematis, proses atau fungsi enkripsi (*E*) dapat dituliskan sebagai: E(M) = C, sedangkan proses atau fungsi dekripsi (*D*) dapat dituliskan sebagai: D(C) = M, dimana M adalah *plaintext* dan C adalah *ciphertext*.

2.3. Pengembangan Kriptografi

Kunci enkripsi dan dekripsi pertamakali digunakan pada tahun 400 SM di Yunani dengan menggunakan Penyandian Transposisi dan Penyandian Substitusi untuk mengubah informasi yang tadinya mudah dibaca menjadi suatu informasi rahasia. Kemudian, di zaman Romawi, kunci enkripsi dan dekripsi mulai menggunakan algoritma matematika untuk

membuat informasi rahasia, yang dinamakan dengan *Caesar Chiper*. Berikutnya, di Inggris, dikenal penggunaan kunci enkripsi dan dekripsi bernama *Playfair Chiper*. Pada saat Perang Dunia I, Jerman menggunakan ADFVGX Chiper sebagai kunci enkripsi dan dekripsi informasi-informasi rahasia yang disampaikan oleh kurir.

Dalam Perang Dunia II, kunci enkripsi dan dekripsi mulai dibuat secara mekanik dengan menggunakan rotor (*machine*). Pada saat itu, Jerman menggunakan Mesin Enigma untuk membuat kunci enkripsi, sedangkan Jepang menggunakan Purple, Inggris menggunakan Typex dan USA menggunakan Sigaba/M-134. Mesin Enigma diperlihatkan dalam Gambar 2.2. berikut.

Gambar 2.2. Mesin Enigma [1]

Seiring meluasnya penggunaan komputer, pada pertengahan tahun 1970-an, Whitfield Diffie dan Martin Hellman menemukan teknik enkripsi asimetris yang merevolusi metode kriptografi. Dengan metode asimetris, kunci publik yang digunakan dapat berupa pasangan kunci kriptografi yang salah satunya digunakan untuk proses enkripsi dan yang lainnya digunakan untuk proses dekripsi, dimana

semua orang dapat menggunakannya untuk mengenkripsi suatu informasi, tetapi hanya satu pihak yang dapat menerjemahkan informasi yang dienkripsi. Contoh metode enkripsi algoritma adalah RSA yang dibuat oleh Rivest, Shamir dan Adleman pada tahun 1977, *Elliptic Curve Cryptosistem* (ECC) serta LUC

Dalam kunci simetris, kunci untuk membuat informasi yang disandikan sama dengan kunci untuk membuka informasi yang disandikan. Jadi, pembuat informasi dan penerimanya harus memiliki kunci yang sama. Siapapun yang memiliki kunci tersebut, termasuk pihak-pihak yang tidak diinginkan, dapat membuat dan membongkar rahasia *ciphertext*. Masalah yang paling jelas disini terkadang bukanlah masalah pengiriman *ciphertext*-nya, melainkan masalah bagaimana menyampaikan kunci simetris tersebut kepada pihak yang diinginkan. Contoh algoritma kunci simetris yang terkenal adalah *Data Encryption Standard (DES*) yang kemudian di standarisasi oleh ANSI menjadi *Data Encryption Algorithm* (DEA). Pada tahun 1990, Xuejia Lai dan James L Massey membuat *International Data Encryption Algorithm (IDEA)*.

2.4. Pengenalan Steganografi

Steganografi merupakan metode menyembunyikan informasi dengan suatu media sehingga seolah-olah terlihat seperti informasi biasa. Media yang digunakan umumnya berbeda dengan media pembawa informasi rahasia, dimana dengan steganografi, informasi menjadi tidak terlihat secara jelas.

Berbagai macam media elektronik kini telah dapat digunakan untuk melakukan berbagai fungsi steganografi

dengan berbagai macam tujuan dan fungsi yang diharapkan oleh penggunanya. Steganografi umumnya digunakan untuk memberi tanda khusus dalam suatu karya (surat berharga) dalam format media elektronik sebagai identifikasi keaslian.

Untuk menyembunyikan informasi perlu adanya dua media. Media pertama adalah media penyamaran seperti citra, suara, video dan sebagainya yang terlihat tidak mencurigakan untuk menyimpan informasi yang dirahasikan. Media kedua adalah informasi yang ingin disembunyikan.

Secara umum, terdapat dua proses dalam steganografi, yaitu proses penanaman (*embedding*) untuk menyisipkan pesan ke dalam media pertama dan proses penerjemahan (*decoding*) untuk menemukan informasi yang disembunyikan. Kedua proses ini memerlukan kunci rahasia yang dinamakan kunci penyembunyian (*stego-key*) agar hanya pihak yang berhak saja yang dapat melakukan penanaman dan penerjemahan informasi. Penyembunyian informasi rahasia ke dalam media digital dapat mengubah kualitas media.

Kriteria yang harus diperhatikan dalam penyembunyian informasi adalah:

1. *Fidelity*

   Mutu citra penyamaran tidak jauh berubah. Setelah penambahan informasi rahasia, citra hasil steganografi masih terlihat dengan baik. Pengamat tidak mengetahui kalau dalam citra tersebut terdapat informasi rahasia.

2. *Robustness*

   Data yang disembunyikan harus tahan terhadap manipulasi yang dilakukan pada citra penyamaran (seperti perubahan kontras, penajaman, pemampatan, penambahan efek, perbesaran gambar, pemotongan). Jika pada citra

dilakukan operasi pengolahan citra, informasi yang disembunyikan tidak rusak.

3. *Recovery*
   Informasi yang disembunyikan harus dapat diungkapkan kembali.

Suatu steganografi memiliki tiga aspek yang menentukan berhasil tidaknya proses steganografi, yaitu:

1. Kapasitas (*capacity*)
   Merupakan jumlah informasi yang dapat disembunyikan dalam media penyamaran.
2. Keamanan (*security*)
   Mewujudkan kerahasiaan informasi yang membuat pengamat tidak mampu mendeteksi informasi tersembunyi.
3. Ketahanan (*robustness*)
   Merupakan jumlah perubahan pada media penyamaran dimana informasi yang disembunyikan dapat bertahan dan tidak rusak.

Terdapat tujuh jenis teknik steganografi, yaitu:

1. Penanaman (*Embedding/Injection*)
   Merupakan teknik steganografi yang menanamkan informasi yang dirahasiakan secara langsung ke suatu media. Salah satu masalah dari teknik ini adalah ukuran media yang diinjeksi menjadi lebih besar daripada ukuran normalnya sehingga mudah dideteksi.
2. Penggantian (*Substitution*)
   Merupakan teknik steganografi yang menggantikan informasi biasa dengan informasi rahasia. Biasanya, hasil teknik ini tidak terlalu mengubah ukuran data asli, tetapi tergantung pada *file* media dan informasi yang

disembunyikan dimana teknik ini dapat menurunkan kualitas media yang ditumpangi.

3. Perpindahan (*Domain Transformation*)
   Merupakan teknik steganografi yang menyembunyikan informasi pada ruang perpindahan.
4. Spektrum Sebaran (*Spread Spectrum*)
   Merupakan teknik steganografi yang menggunakan *pseudo-noise code*, yang independen terhadap data informasi sebagai modulator bentuk gelombang untuk menyebarkan energi sinyal dalam suatu jalur komunikasi (bandwith) yang lebih besar daripada sinyal jalur komunikasi informasi. Oleh penerima, sinyal dikumpulkan kembali menggunakan replika *pseudo-noise code* tersinkronisasi.
5. Metode Statistik (*Statistical Method*)
   Merupakan teknik steganografi yang menanamkan satu bit informasi pada media penyamaran dan mengubah statistik walaupun hanya 1 bit. Perubahan statistik ditunjukkan dengan indikasi 1 dan jika tidak ada perubahan, terlihat indikasi 0. Sistem ini bekerja berdasarkan pada kemampuan penerima dalam membedakan antara informasi yang dimodifikasi dan yang belum dimodifikasi.
6. Perbedaan (*Distortion*)
   Merupakan teknik steganografi yang menciptakan perubahan atas citra yang ditumpangi oleh informasi yang dirahasiakan.
7. Pembuatan Penutup (*Cover Generation*)
   Merupakan teknik steganografi yang menggunakan penutup untuk menyembunyikan informasi.

2.5. Pengembangan Steganografi

Steganografi telah mulai digunakan oleh penguasa Yunani untuk mengirim pesan rahasia dengan menggunakan kepala prajurit sebagai media. Dalam hal ini, rambut prajurit dibotaki, lalu pesan rahasia ditulis pada kulit kepala prajurit. Ketika rambut prajurit tumbuh, prajurit tersebut diutus untuk membawa pesan rahasia di balik rambutnya. Selain itu, masyarakat Yunani kuno telah mengenal penggunaan lilin sebagai media penyembunyi pesan. Pesan ditulis pada suatu lembaran, dan lembaran tersebut ditutup dengan lilin untuk menyembunyikan pesan tersebut. Pihak penerima kemudian menghilangkan lilin pada lembaran tersebut untuk melihat pesan yang disampaikan oleh pihak pengirim.

Masih di zaman Yunani, Herodotus menggunakan steganografi dalam tulisannya untuk menyembunyikan pesan rahasia. Namun, teknik steganografi pada waktu tersebut masih sangat sederhana, yaitu dengan menggunakan kode yang sangat mudah dipecahkan.

Di zaman Romawi, pembuatan steganografi menggunakan tinta tidak terlihat (invisible ink) yang dibuat dari campuran sari buah, susu, dan cuka untuk menuliskan pesan. Tulisan di atas kertas dapat dibaca dengan memanaskan kertas tersebut.

Pada abad pertengahan, steganografi berkembang lebih kompleks dan mulai digunakan untuk tujuan militer. Salah satu teknik yang populer pada abad pertengahan adalah menggunakan jaring laba-laba sebagai media menyembunyikan pesan rahasia. Selain itu, teknik penyembunyian pesan dalam ilustrasi dan kaligrafi juga menjadi populer pada abad pertengahan.

Dalam revolusi industri, steganografi semakin berkembang dengan penggunaan telegraf dan telepon. Teknik steganografi pada masa ini umumnya menggunakan kode Morse dan nada telepon sebagai media penyembunyian pesan rahasia.

Pada abad ke-20, seiring dengan perkembangan teknologi komputer, perkembangan steganografi menjadi lebih kompleks dan mudah dilakukan. Pada tahun 1985, Neil F. Johnson dan Sushil Jajodia menulis artikel ilmiah yang membahas pengembangan teknik steganografi modern berbasis komputer menggunakan metode penyisipan bit pada file digital. Teknik ini kemudian dikembangkan dan semakin populer digunakan dalam komunikasi digital. Pada masa sekarang, steganografi semakin banyak digunakan dalam berbagai aplikasi, seperti pengiriman pesan rahasia, perlindungan hak cipta, dan pengamanan informasi rahasia.

Teknik steganografi modern yang banyak digunakan berbasis komputer adalah teknik penyisipan bit dalam file digital. Dalam teknik ini, pesan rahasia disisipkan ke dalam file digital, seperti gambar atau audio, dengan cara memodifikasi bit-bit yang tidak terlihat oleh mata manusia. Bit-bit yang dimodifikasi tidak akan mempengaruhi kualitas file digital dan tidak akan terlihat oleh orang lain yang tidak mengetahui teknik steganografi. Salah satu teknik steganografi modern yang sangat populer adalah teknik Least Significant Bit (LSB). Dalam teknik ini, pesan rahasia disisipkan ke dalam file digital dengan cara mengganti bit-bit paling tidak signifikan dalam file digital dengan bit-bit pesan rahasia. Bit-bit yang diganti tidak akan mempengaruhi kualitas file digital dan tidak akan terlihat oleh mata manusia.

Meskipun steganografi berguna dalam pengiriman pesan

rahasia, perlindungan hak cipta, dan pengamanan informasi rahasia, steganografi juga dapat digunakan untuk melakukan kejahatan dan aktivitas terorisme. Selain itu, terdapat kemungkinan steganografi dimanfaatkan untuk menghindari deteksi perangkat lunak keamanan dan sarana melakukan serangan siber. Oleh karena itu, pihak yang terlibat dalam keamanan dan penegakan hukum perlu memahami teknik steganografi dan dapat mengidentifikasi pesan rahasia yang disembunyikan dalam file digital.

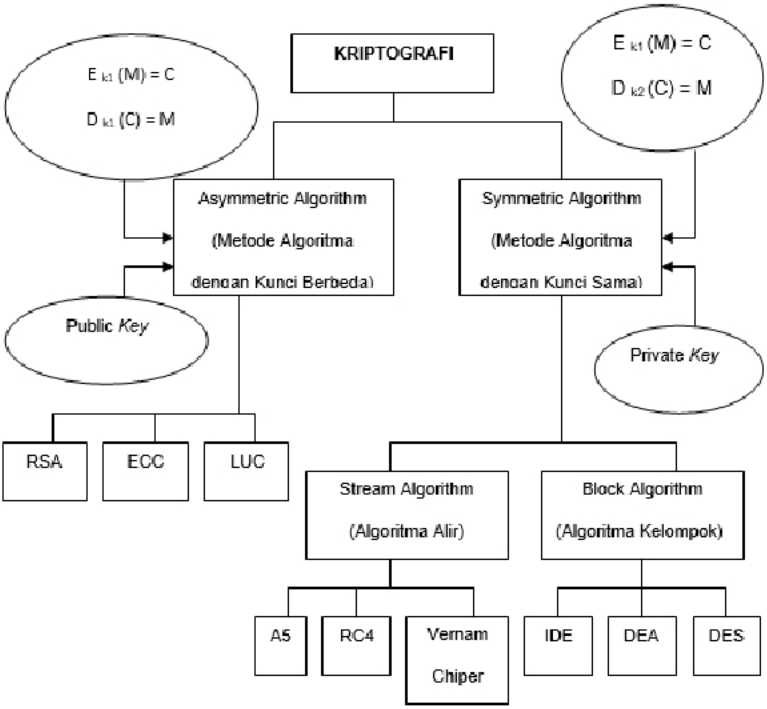
Pada masa sekarang, steganografi digunakan untuk tujuan legitimasi seperti menggunakan citra dengan digital watermarking atau fingerprinting untuk melindungi hak cipta. Steganografi juga digunakan sebagai tag-notes untuk citra online.

Di masa depan, steganografi masih akan terus berkembang seiring dengan kemajuan teknologi digital. Perkembangan teknologi digital seperti Internet of Things (IoT) dan teknologi 5G akan membuka lebih banyak peluang untuk penggunaan steganografi dalam komunikasi digital. Oleh karena itu, perlu ada upaya untuk meningkatkan kesadaran dan pemahaman tentang steganografi di kalangan masyarakat.

BAB III
KUNCI ENKRIPSI DAN DEKRIPSI

3.1. Klasifikasi Kunci Enkripsi dan Dekripsi

Klasifikasi kunci enkripsi dan dekripsi diperlihatkan dalam Gambar 3.1. berikut.



Gambar 3.1. Klasifikasi Kunci Enkripsi dan Dekripsi

Kunci enkripsi dan dekripsi dapat digolongkan atas dua

algoritma dasar untuk pembuatan kunci enkripsi dekripsi, yaitu:

1. Kunci enkripsi dan dekripsi simetris, yaitu kunci enkripsi (K_1) dan kunci dekripsi (K_2) menggunakan algoritma yang sama, dimana nilai $K_1 = K_2 = K$. Enkripsi simetris atau kunci simetris disebut juga dengan nama algoritma kunci rahasia (*secret key algorithm*) atau algoritma kunci tunggal (*single key algorithm*). Karena kunci enkripsi dan dekripsinya sama, kunci simetris mensyaratkan kunci K harus dirahasiakan dari pihak-pihak yang tidak berkepentingan, sehingga disebut juga kunci pribadi (*private key*).
2. Kunci enkripsi dan dekripsi asimetris, yaitu kunci enkripsi (K_1) dan kunci dekripsi (K_2) menggunakan algoritma yang sama, dimana nilai $K_1 \neq K_2$. Kunci enkripsi ini dikenal dengan nama kunci publik (*public key*), karena dalam algoritma ini, kunci enkripsi K_1 tidak perlu dirahasiakan dan dapat diumumkan kepada semua orang, sedangkan kunci dekripsi K_2 harus dijaga kerahasiaannya.

Kunci enkripsi dan dekripsi simetris dapat digolongkan lagi menjadi dua algoritma dasar, yaitu:

1. *Stream cipher algorithm,* dimana setiap bit dari informasi dienkripsi secara berurutan dengan menggunakan 1 bit kunci (melakukan enkripsi terhadap semua bit). Contoh: Vernam *cipher*.
2. *Block cipher algorithm,* dimana enkripsi dilakukan terhadap kelompok-kelompok bit informasi dengan ukuran tertentu. Contoh: *Data Encyption Standard (DES).*

3.2. Enkripsi Dekripsi Simetris

Kunci enkripsi dan dekripsi simetris adalah enkripsi dengan kunci enkripsi yang sama dengan kunci dekripsi, dimana nilai $K_1 = K_2 = K$. Enkripsi simetris atau kunci simetris disebut juga dengan nama algoritma kunci rahasia (*secret key algorithm*) atau algoritma kunci tunggal (*single key algorithm*). Karena kunci enkripsi dan dekripsinya sama, kunci simetris mensyaratkan kunci K harus dirahasiakan dari pihak-pihak yang tidak berkepentingan.

Tingkat keamanan kriptografi yang menggunakan algoritma ini ditentukan oleh kerahasiaan kunci K yang digunakan. Jika seseorang hendak mengirimkan suatu informasi rahasia kepada orang lain, atau melakukan *secure communication*, orang tersebut harus terlebih dahulu memberitahu kunci K yang hendak digunakan kepada pihak yang dituju. Hal ini jelas membutuhkan suatu saluran komunikasi yang benar-benar aman dan tidak dapat disadap (*secure channel*).

Secara matematis, algoritma enkripsi simetris dapat dituliskan sebagai berikut:

$$e_K (M) = C \text{ diubah menjadi } d_K (C) = M$$

Contoh kunci simetris ini adalah *Data Encryption Standard* (DES), Lucifer, IDEA, FEAL, AS, KPD, LOKI dan NewDES. Standar kripografi dengan kunci simetris saat ini adalah DES dan IDEA.

Algoritma *International Data Encryption Algorithm (IDEA)* merupakan salah satu contoh kunci simetris yang dibuat pertamakali pada tahun 1990 oleh Xuejia Lai dan James L Massey. Algoritma IDEA merupakan kunci simetris yang beroperasi pada suatu blok informasi terbuka 64 bit. Kunci

yang digunakan dalam proses enkripsi dan dekripsi berukuran sama, yaitu 2 x 64 = 128 bit. Informasi rahasia yang dihasilkan juga berupa blok dengan ukuran 64 bit. Proses dekripsi menggunakan blok penyandi yang sama dengan blok proses enkripsi dimana kunci dekripsinya diturunkan dari kunci enkripsi. Algoritma ini menggunakan operasi campuran dari operasi aljabar yang berbeda, yaitu operasi XOR dan operasi dengan penjumlahan modulo $2^{16} + 1$.

Pada proses enkripsi Algoritma IDEA, terdapat tiga operasi yang berbeda untuk pasangan subblok 16 bit yang digunakan, sebagai berikut:

1. XOR dua subblok 16 bit per bit,
2. Jumlahkan bilangan bulat modulo 2^{16} dua sub blok 16 bit di mana kedua sub blok ini dianggap sebagai representasi biner dari bilangan bulat biasa,
3. Kalikan bilangan bulat modulo $2^{16} + 1$ dua subblok 16 bit, di mana kedua subblok 16 bit itu dianggap sebagai representasi biner dari bilangan bulat biasa kecuali subblok nol dianggap mewakili bilangan bulat 2^{16}.

Blok informasi terbuka 64 bit, X, dibagi menjadi 4 subblok 16 bit, X1, X2, X3, X4, sehingga X = (X1, X2, X3, X4). Keempat subblok 16 bit itu ditrnsformasikan menjadi 4 sublok 16 bit, Y1, Y2, Y3, Y4, sebagai informasi rahasia 64 bit Y = (Y1, Y2, Y3, Y4) yang berada di bawah kendali 52 subblok kunci 16 bit yang dibentuk dari blok kunci 128 bit.

Keempat subblok 16 bit, X1, X2, X3, X4 digunakan sebagai masukan untuk putaran pertama dari algoritma IDEA. Dalam setiap putaran dilakukan operasi XOR, penjumlahan, perkalian antara dua subblok 16 bit dan diikuti pertukaran antara subblok 16 bit keluaran kedua dan ketiga. Keluaran

putaran sebelumnya menjadi masukan putaran berikutnya. Setelah putaran ke delapan dilakukan transformasi keluaran yang dikendalikan oleh 4 subblok kunci 16 bit. Pada setiap putaran dilakukan serangkaian operasi berikut:

1. Kalikan X1 dengan sub kunci pertama,
2. Jumlahkan X2 dengan sub kunci kedua,
3. Jumlahkan X3 dengan sub kunci ketiga,
4. Kalikan X4 dengan sub kunci keempat,
5. Operasi XOR hasil langkah (1) dan (3),
6. Operasi XOR hasil langkah (2) dan (4),
7. Kalikan hasil langkah (5) dengan sub kunci kelima,
8. Jumlahkan hasil langkah (6) dan (7),
9. Kalikan hasil langkah (8) dengan sub kunci keenam,
10. Jumlahkan hasil langkah (7) dan (9),
11. Operasi XOR hasil langkah (1) dan (9),
12. Operasi XOR hasil langkah (3) dan (9),
13. Operasi XOR hasil langkah (2) dan (10),
14. Operasi XOR hasil langkah (4) dan (10).

Keluaran setiap putaran adalah 4 subblok yang dihasilkan pada langkah (11) sampai dengan (14) dan menjadi masukan pada putaran berikutnya. Setelah putaran kedelapan, terdapat transformasi keluaran, yaitu:

1. Kalikan X1 dengan sub kunci pertama
2. Jumlahkan X2 dengan sub kunci ketiga
3. Jumlahkan X3 dengan sub kunci kedua.
4. Kalikan X4 dengan sub kunci keempat
5. Terakhir, keempat subblok 16 bit yang merupakan hasil operasi (1) sampai dengan (4) digabung kembali menjadi blok informasi rahasia 64 bit.

Proses dekripsi menggunakan algoritma yang sama

dengan proses enkripsi, tetapi 52 subblok kunci yang digunakan masing-masing merupakan hasil turunan 52 subblok kunci enkripsi.

3.3. Enkripsi Dekripsi Asimetris

Kunci enkripsi dan dekripsi asimetris adalah enkripsi dengan kunci enkripsi K_1 yang berbeda dengan kunci dekripsi K_2. Kunci enkripsi ini dikenal dengan nama algoritma kunci publik, karena dalam algoritma ini, kunci enkripsi K_1 tidak perlu dirahasiakan dan dapat diumumkan kepada semua orang, sedangkan kunci dekripsi K_2 harus dijaga kerahasiaannya. Dalam hal ini, kunci dekripsi K_2 tidak dapat dihitung atau diturunkan dari kunci enkripsi K_1. Seseorang yang tidak dikenal sekalipun dapat menggunakan kunci enkripsi K_1 untuk menyandikan informasi, tetapi hanya si penerima yang sah saja yang memiliki kunci dekripsi K_2.

Secara matematis, algoritma enkripsi asimetris dapat dituliskan sebagai berikut:

$$e_{K1}(M) = C \text{ diubah menjadi } d_{K2}(C) = M$$

Kunci enkripsi yang dipublikasikan secara umum disebut dengan kunci publik (*public key*), sedangkan kunci dekripsi yang harus dijaga kerahasiaannya disebut dengan kunci pribadi (*private key*). Beberapa contoh algoritma asimetris adalah algoritma ElGamal, Rabin, *Elliptic Curve Cryptosistem* (ECC), Diffie Helman, LUC, dan Rivest Shamir Addleman (RSA).

Salah satu contoh kunci asimetris adalah kunci enkripsi yang dibuat dengan algoritma Rivest Shamir Addleman (RSA) yang ditemukan pertamakali pada tahun 1977 oleh tiga orang peneliti, yaitu R.L Rivest, A. Shamir dan L Addleman. Algoritma ini dibuat berdasarkan fakta bahwa dalam

perhitungan dengan menggunakan perangkat, untuk menemukan suatu bilangan prima yang besar sangat mudah, namun untuk mencari faktor dari perkalian dua bilangan prima yang besar sangatlah sulit, bahkan hampir tidak mungkin. Algoritma tersebut dipatenkan oleh Massachusetts Institute of Technology pada tahun 1983 di Amerika Serikat sebagai U.S. Patent 4405829. Paten tersebut berlaku hingga 21 September 2000. Sejak Algoritma RSA dipublikasikan sebagai aplikasi paten, regulasi di sebagian besar negara-negara lain tidak memungkinkan penggunaan paten tersebut.

RSA merupakan salah satu contoh algoritma pada enkripsi *public key*. RSA merupakan algoritma yang cocok untuk *digital signature* seperti halnya enkripsi, dan salah satu enkripsi yang paling maju dalam bidang kriptografi adalah *public key*. RSA masih digunakan secara luas dalam protokol *electronic commerce*, dan dipercaya dalam mengamankan informasi dengan menggunakan kunci yang cukup panjang.

RSA merupakan salah satu algoritma kunci publik yang menggunakan penyandian blok, yaitu penyandian dengan proses perhitungan enkripsi/dekripsi yang dilakukan dengan hitungan perblok. Panjang setiap blok harus memenuhi syarat-syarat tertentu, agar proses perhitungan tidak sesuai dengan yang sebenarnya.

Standar internasional yang mengatur panjang blok serta proses *padding* (penambahan karakter agar mencapai panjang yang sesuai) yang diijinkan dalam algoritma RSA adalah *Public Key Cryptography Standards* (PKCS). Susunan algoritma RSA terdiri dari:

1. Proses Enkripsi, dengan rumus $C = M^e \bmod n$.
2. Proses Dekripsi, dengan rumus $M = C^d \bmod n$.

dimana:

M = bilangan bulat yang menyatakan informasi yang hendak diubah (disandikan)

C = bilangan bulat yang menyatakan informasi yang telah diubah.

e = kunci enkripsi

d = kunci dekripsi

n = modulus (publik)

Dalam aplikasinya, bilangan-bilangan bulat e dan n yang merupakan kunci publik RSA diletakkan pada suatu direktori publik yang dapat diakses semua orang, sehingga setiap orang dapat mengirimkan informasi tersandi ke si pengelola kunci publik e dan n tersebut.

Apabila seorang pengirim informasi berkeinginan untuk mengijinkan seseorang mengirimkan suatu informasi pribadi (*private message*) kepadanya melalui media transmisi yang tidak aman (*insecure*), maka diperlukan pembuatan pasangan kunci *public key* dan *private key*, sebagai berikut:

1. Pilih dua bilangan prima $p \neq q$ secara acak dan terpisah untuk tiap p dan q.
2. Hitung modulus n = p q.
3. Kemudian n hasil perkalian dari p dikalikan dengan q.
4. Hitung fungsi Euler's Totient $\varphi(n) = (p-1)(q-1)$.
5. Pilih bilangan bulat (bilangan bulat) antara satu dan φ ($1 < e < \varphi$) yang juga merupakan *coprime* dari φ.
6. Hitung nilai bilangan bulat *d* di mana $1<d<\varphi(n)$ sehingga $d = e^{-1} \mod \varphi(n)$ atau $e.d = 1 \pmod{\varphi(n)}$ dengan menggunakan Algoritma Euclidean yang diperluas.
7. *Public key* dari sistem ini adalah n dan e sedangkan *private key*-nya adalah d.

8. Selanjutnya, kunci e dan n ini diletakkan pada suatu direktori publik yang dapat diakses oleh semua orang, sedangkan kunci d haruslah tetap dijaga kerahasiaannya. Dengan demikian, semua orang dapat mengenkripsikan informasi yang ditujukan kepada pemilik kunci tersebut, namun hanya pemilik kunci dekripsii yang dapat mendekripsikan informasi yang disandikan.

Bilangan prima dapat diuji probabilitasnya menggunakan *Fermat's little theorem* di mana a^(n-1) mod n = 1 jika n adalah bilangan prima. Pengujian dilakukan dengan beberapa nilai a yang menghasilkan kemungkinan yang tinggi bahwa n adalah bilangan prima. *Carmichael numbers* (angka-angka Carmichael) diuji pada seluruh nilai a. Pengirim informasi mengirimkan *public key* kepada penerima, dan tetap merahasiakan *private key* yang digunakan. Nilai *p* dan *q* sangat sensitif karena merupakan faktorial dari *N*, dan membuat perhitungan dari *d* menghasilkan *e*. Jika *p* dan *q* tidak disimpan dari *private key*, maka *p* dan *q* telah terhapus bersama nilai-nilai lain dari proses pembuatan kunci.

Pembuatan enkripsi informasi dapat dilakukan sebagai berikut:

1. Pengirim mengirimkan informasi *m* ke penerima.
2. Penerima mengubah informasi *m* menjadi angka $n < N$, menggunakan protokol yang sebelumnya telah disepakati dan dikenal sebagai *padding scheme*.
3. Dengan memiliki *n* dan mengetahui *N* dan *e*, yang telah diumumkan oleh pengirim, penerima kemudian menghitung *ciphertext c* yang terkait pada *n*, dengan rumus:

$$c = n^e \mod N$$

4. Perhitungan tersebut dapat diselesaikan dengan cepat menggunakan metode *exponentiation by squaring*. Penerima kemudian mengirimkan *c* kepada pengirim.

Pembuatan dekripsi informasi dapat dilakukan sebagai berikut:

1. Pengirim menerima *c* dari penerima dan mengetahui *private key* yang digunakan oleh pengirim.sendiri.
2. Pengirim kemudian memulihkan *n* dari *c* dengan rumus berikut:

$$n = c^d \mod N$$

3. Perhitungan diatas menghasilkan *n*, sehingga pengirim dapat mengembalikan informasi semula 4. Prosedur dekripsi bekerja karena

$$c^d \equiv (n^e)^d \equiv n^{ed} \pmod N.$$

4. Dikarenakan $ed \equiv 1 \pmod{p-1}$ dan $ed \equiv 1 \pmod{q-1}$, maka hasil dari *Fermat's little theorem*.adalah:

$$n^{ed} \equiv n \pmod p$$

dan

$$n^{ed} \equiv n \pmod q$$

5. Dikarenakan *p* dan *q* merupakan bilangan prima yang berbeda, aplikasi *Chinese remainder theorem* akan menghasilkan dua macam kongruen

$$n^{ed} \equiv n \pmod{pq}.$$

dan

$$c^d \equiv n \pmod N.$$

Padding Scheme dibangun secara hati-hati untuk menghindari munculnya nilai dari *m* yang menyebabkan masalah keamanan. Pada kenyataannya, sistem yang

menggunakan nilai *e* kecil, seluruh karakter tunggal ASCII pada informasi disandikan menggunakan skema yang tidak aman, dikarenakan nilai terbesar *n* adalah nilai 255, dan 255^3 akan menghasilkan nilai yang lebih kecil dari modulus yang sewajarnya. Proses dekripsi menjadi masalah yang sederhana dengan mengambil pola dasar dari *ciphertext* tanpa perlu menggunakan modulus *N*. Sebagai konsekuensinya, standar seperti PKCS dirancang dengan sangat hati-hati sehingga membuat informasi dapat terenkripsi secara aman. P*adding scheme* merupakan hal yang esensial untuk mengamankan pengesahan informasi seperti halnya pada enkripsi informasi, oleh karena itu kunci yang sama tidak digunakan pada proses enkripsi dan pengesahan.

Untuk menemukan bilangan prima besar *p* dan *q* biasanya dengan mencoba serangkaian bilangan acak dengan ukuran yang tepat menggunakan probabilitas bilangan prima yang dapat dengan cepat menghapus hampir semua bilangan bukan prima. Nilai *p* dan *q* seharusnya tidak "saling-berdekatan", agar faktorisasi fermat pada *N* berhasil. Selain itu, jika *p*-1 atau *q*-1 memiliki faktorisasi bilangan prima yang kecil, *N* dapat difaktorkan secara mudah dan nilai-nilai dari *p* atau *q* dapat diacuhkan.

Tingkat keamanan penyandian RSA berdasarkan pada kompleksitas bilangan yang digunakan sebagai kunci e,n dan d. Semakin besar dan kompleks bilangan yang digunakan, maka tingkat keamanan yang diperoleh juga semakin tinggi. Meskipun demikian, masalahnya, penggunaan kunci yang semakin kompleks membuat kecepatan akses menjadi lebih rendah, sehingga diperlukan keseimbangan (*trade off*) untuk memperoleh kunci yang optimal.

Penyerangan yang paling umum pada RSA adalah pada penanganan masalah faktorisasi bilangan yang sangat besar. Apabila terdapat metode faktorisasi yang baru dan cepat telah dikembangkan, maka ada kemungkinan untuk membongkar RSA. Pada tahun 2005, bilangan faktorisasi terbesar yang digunakan secara umum adalah sepanjang 663 bit, menggunakan metode distribusi mutakhir. Beberapa pakar meyakini bahwa kunci 1024-bit ada kemungkinan dipecahkan pada waktu dekat, tetapi tidak ada seorangpun yang berpendapat kunci 2048-bit akan dipecahkan di masa depan.

Apabila seseorang mendapatkan *public key* N dan e, dan *ciphertext* c, belum tentu ia dapat secara langsung memperoleh d yang dijaga kerahasiannya oleh pengirim. Permasalahan untuk menemukan n seperti pada $n^e = c \bmod N$ dikenal sebagai permasalahan RSA. Cara yang paling efektif dilakukan oleh kriptanalis untuk memperoleh n dari c adalah dengan melakukan faktorisasi N ke dalam p dan q, dengan tujuan untuk menghitung $(p-1)(q-1)$ yang dapat menghasilkan d dari e. Tidak ada metode waktu polinomial untuk melakukan faktorisasi pada bilangan bulat berukuran besar di komputer saat ini, tapi hal tersebut pun masih belum terbukti. Jika N sepanjang 256-bit atau lebih pendek, N akan dapat difaktorisasi dalam beberapa jam pada komputer, dengan menggunakan perangkat lunak yang tersedia secara bebas. Jika N sepanjang 512-bit atau lebih pendek, N akan dapat difaktorisasi dalam hitungan ratusan jam seperti pada tahun 1999. Secara teori, perangkat keras bernama TWIRL dan penjelasan dari Shamir dan Tromer pada tahun 2003 mengundang berbagai pertanyaan akan keamanan dari kunci 1024-bit. Sangat disarankan bahwa N setidaknya sepanjang 2048-bit. Pada thaun 1993, Peter Shor

menerbitkan Algoritma Shor, yang dapat menunjukkan bahwa suatu komputer quantum secara prinsip dapat melakukan faktorisasi dalam waktu polinomial, mengurai RSA dan algoritma lainnya.

Sebagaimana halnya *cipher*, pendistribusian *public key* RSA menjadi hal yang sangat penting dalam keamanan. Distribusi kunci harus aman dari *man-in-the-middle attack* (penghadang-ditengah-jalan). Pengamanan terhadap ancaman keamanan semacam ini yaitu menggunakan sertifikat digital atau komponen lain dari infrastuktur *public key*.

3.4. Perbandingan Kunci Simetris Dan Kunci Asimetris

Apabila enkripsi simetris dibandingkan dengan enkripsi asimetris, terdapat kelebihan dan kekurangan pada masing-masing kunci enkripsi tersebut. Pada enkripsi simetris, terdapat kelebihan dalam kecepatan proses dan kecepatan kirim, namun memiliki kelemahan dalam tingkat keamanannya. Kecepatan proses enkripsi dan dekripsi diperoleh melalui efisiensi yang terjadi pada pembuatan kunci. Hal ini membuat kunci simetris dapat digunakan dalam sistem secara *real time*, seperti pada saluran telepon digital.

Pada kunci simetris, seperti kunci DES, membutuhkan jalur komunikasi yang benar-benar aman untuk mengirimkan kunci yang diperlukan sampai ke tempat tujuan. Banyak kemungkinannya, kunci enkripsi dan sekaligus kunci dekripsi dapat disadap oleh orang yang tidak berhak menerimanya. Perlu adanya kesepakatan untuk mengetahui jalur yang khusus untuk kunci. Hal ini akan menimbulkan masalah yang baru karena tidak mudah untuk menentukan jalur yang aman untuk kunci. Permasalahan ini sering disebut dengan *"Key*

Distribution Problem". Selain itu, karena pengguna memerlukan kunci yang berbeda untuk setiap pasang, maka sangat sulit untuk menyimpan dan mengingat kunci yang banyak secara aman. Hal ini menimbulkan kesulitan dalam pengelolaan kunci. Apabila kunci enkripsi dengan dekripsi sampai hilang maka dapat ditebak kriptografi menjadi tidak aman lagi.

Enkripsi asimetris memiliki tingkat keamanan yang relatif lebih baik, namun memiliki kelemahan dalam hal kecepatan prosesnya yang rendah. Pada enkripsi asimetris, seperti kunci RSA, memiliki kecepatan enkripsi 1000 sampai dengan 10000 kali lebih lambat dibandingkan enkripsi simetris, seperti kunci DES. Hal ini dikarenakan kunci enkripsi asimetris dibuat sangat kompleks untuk tujuan pengiriman informasi yang jauh lebih aman. Pada kunci enkripsi asimetris terdapat dua kunci yang digunakan untuk dapat saling berhubungan, yaitu suatu kunci publik (*public key*) dan suatu kunci pribadi (*private key*). Misalnya, pihak A dapat memberikan kunci publik kepada siapa saja yang ingin menerima informasi terenkripsi. Kunci tersebut hanya dapat mengkodekan informasi; tetapi tidak dapat menerjemahkan informasi yang sudah dikodekan. Kunci pribadi harus tetap terjaga dengan aman oleh pihak A. Saat pihak selain A hendak mengirimkan informasi terenkripsi pada pihak A, pihak selain A tersebut dapat mengenkripsinya menggunakan kunci publik A. Ketika A menerima *chipertext* tersebut, maka pihak A akan mendekripsikannya dengan menggunakan kunci pribadi yang dimiliki pihak A. Enkripsi asimetris menambahkan tingkat keamanan pada informasi pihak A, tetapi berakibat pada banyaknya waktu komputasi yang dibutuhkan, sehingga

prosesnya menjadi sangat panjang dan memerlukan waktu yang lebih lama.

3.5. Enkripsi Dekripsi Aliran (Stream Chiper)

Kunci enkripsi dan dekripsi Aliran (*Stream Chiper*) merupakan jenis kunci enkripsi dekripsi simetris yang menghasilkan suatu *keystream* (barisan bit yang digunakan sebagai kunci). Konsentrasi dalam *streaming chiper* umumnya berkaitan dengan sifat sifat teoritik yang menarik dari *one-time pad*, seperti Vernam *cipher* yang menggunakan suatu *string* dari bit yang dihasilkan secara acak. *Keystream* memiliki panjang yang sama dengan pesan *plaintext* dan digabungkan dengan *plaintext* menggunakan *bitwise XOR* untuk menghasilkan *ciphertext*. Karena *keystream* seluruhnya adalah acak, walaupun dengan sumber daya komputasi tak terbatas, seseorang hanya dapat menduga *plaintext* jika dia melihat *ciphertext*. Metode enkripsi dekripsi seperti ini dapat memberikan kerahasiaan yang sempurna (*perfect secrecy*).

Metode *streaming chiper* yang umum digunakan adalah RC4, yaitu salah satu jenis kunci yang mempunyai S-Box, S0, S1,… S255, yang berisi permutasi dari bilangan 0 sampai 255. Pada algoritma enkripsi ini akan membangkitkan *pseudorandom byte* dari *keystream* yang mengalami operasi XOR terhadap *plaintext* untuk menghasilkan *ciphertext*. Untuk menghasilkan *plaintext* semula, maka *ciphertext*-nya akan dikenakan operasi XOR terhadap *pseudorandom byte*nya. Algoritma RC4 menggunakan dua indeks yaitu i dan j. Indeks i digunakan untuk memastikan bahwa suatu elemen berubah, sedangkan indeks j akan memastikan bahwa suatu elemen berubah secara acak.

Secara garis besar, algoritma metode RC4 terbagi menjadi dua bagian, yaitu: *key setup* dan *stream generation*. Pada *key setup* terdapat tiga tahapan proses di dalamnya, yaitu pengenalan S-Box, penyimpanan *key setup* dalam *Key Byte Array*, permutasi pada S-Box. Sedangkan pada *stream generation,* nilai *pseudorandom* yang akan dikenakan operasi XOR menghasilkan *ciphertext* ataupun sebaliknya, menghasilkan *plaintext*.

3.6. Enkripsi Dekripsi Kelompok (Block Chiper)

Kunci enkripsi dan dekripsi Kelompok (*Block Chiper*) adalah algoritma kriptografi yang bekerja pada suatu data berbentuk blok/kelompok data dengan panjang data tertentu (dalam beberapa *byte*), sehingga dalam sekali proses enkripsi atau dekripsi, data yang masuk mempunyai ukuran yang sama. Dalam algoritma penyandian blok (*block cipher*), *plaintext* yang masuk akan diproses dengan panjang blok yang tetap yaitu n, namun terkadang jika ukuran data ini terlalu panjang maka dilakukan pemecahan dalam bentuk blok yang lebih kecil. Jika dalam pemecahan dihasilkan blok data yang kurang daripada jumlah data dalam blok maka akan dilakukan proses *pading* (penambahan beberapa bit).

Beberapa contoh *block cipher* adalah:

1. *Data Encryption Standard (DES)*
   Merupakan algoritma *block cipher* dengan ukuran blok data 64 bit dan ukuran kunci 56 bit. DES untuk saat ini sudah dianggap tidak aman lagi karena ukuran kuncinya yang sangat pendek 56 bit.
2. Skema Global DES
   Pada awalnya, blok *plaintext* dipermutasi dengan matriks

permutasi awal (*initial permutation* atau IP). Hasil permutasi awal tersebut kemudian dienkripsi sebanyak 16 kali atau 16 putaran. Setiap putarannya menggunakan kunci dalam yang berbeda. Hasil proses enkripsi kembali dipermutasi dengan matriks permutasi balikan (invers initial permutation atau IP-1) menjadi blok *ciphertext*.

3. Triple DES
   Merupakan variasi *Data Encryption Standard (DES)* yang menggunakan kunci 64 bit terdiri atas 56 bit kunci yang efektif dan 8 bit paritas. Bit paritas/bit pemeriksa yaitu bit tambahan yang ditempatkan di posisi akhir suatu *byte* yang berguna untuk memeriksa kesalahan selama transmisi. Ukuran setiap blok Triple DES adalah 8 *byte* dengan menggunakan tiga kunci yang berbeda.
4. *Advanced Encryption Standard* (AES)
   Merupakan *symmetric key encryption* untuk data elektonik. Pertamakali digunakan oleh Pemerintah USA dan sekarang digunakan oleh seluruh dunia menggantikan DES. Standar AES terdiri dari 128, 192 dan 256 bit.
5. Carlisle Adams dan Stafford Tavares (CAST)
   Dibuat oleh Carlisle Adams dan Stafford Tavares. CAST populer dengan 64-bit blok *cipher* yang termasuk kelas algoritma enkripsi dikenal sebagai *cipher* Feistel.
6. Serpent
   Merupakan *block cipher* yang sangat cepat dan cukup aman yang dikembangkan oleh Ross Anderson, Eli Biham dan Lars Knudsen. Serpent dapat bekerja dengan kombinasi panjang kunci yang berbeda.
7. Blowfish
   Merupakan algoritma enkripsi simetris yang dirancang

pada tahun 1993 oleh Bruce Schneier sebagai alternatif untuk algoritma enkripsi yang ada. Blowfish memiliki ukuran 64-bit blok dan panjang kunci variabel dari 32 bit sampai 448 bit.

8. Twofish
   Merupakan *block cipher* simetris yang memiliki blok data 128 bit dan menerima kunci yang panjangnya dapat mencapai 258 bit. Twofish dirancang oleh Bruce Schneier, John Kelsey, Chris Hall, Niels Ferguson, Davis Wagner dan Doug Whiting.
9. *International Data Encryption Algorithm* (IDEA)
   Merupakan kunci simetris yang dikembangkan oleh Dr. Xu. Lai dan Profesor J. Massey untuk menggantikan DES meskipun masih menggunakan kunci 128 bit.

3.7. Mesin Enkripsi Dekripsi

Mesin yang digunakan untuk membuat kunci enkripsi dan dekripsi, di antaranya adalah Mesin Enigma, yaitu suatu mesin rotor elektromekanik yang digunakan untuk mengenkripsi informasi dan mendekripsikan kembali informasi tersebut. Mesin Enigma ditemukan oleh seorang Jerman bernama Arthur Scherbius di Berlin pada tahun 1918, dan digunakan oleh militer dan pemerintah Jerman Nazi sebelum dan selama Perang Dunia II. Terdapat beberapa versi Mesin Enigma dan yang terkenal adalah yang dipakai oleh Wehrmacht (angkatan bersenjata Jeman Nazi) sejak tahun 1928. Pada awalnya, Nazi menganggap bahwa Mesin Enigma adalah mesin kriptografi teraman di dunia. Namun pihak sekutu dapat memecahkan kode *ciphertext* yang dihasilkan oleh Mesin Enigma pada tahun 1932. Metode dekripsi untuk

mesin ini ditemukan oleh tim matematikawan muda yang terdiri dari Marian Rejewski, Jerzy Rozycki, dan Henryk Zygalski. Pihak Nazi yang menyadari hal ini pun mendesain ulang Mesin Enigma pada tahun 1939, sehingga metode tersebut tidak dapat digunakan kembali. Meskipun demikian, berdasarkan metode dari Polandia, Britania dan Perancis berhasil membuat mesin pemecah kode untuk Mesin Enigma yang baru ini dan terbukti menjadi faktor penting kemenangan Sekutu pada Perang Dunia II, dan berhasil memperpendek lamanya Perang Dunia II.

Desain Mesin Enigma versi militer yang banyak dipakai oleh angkatan bersenjata Jerman Nazi terdiri dari: papan ketuk, lampu, papan steker, pengacak (*scrambler*), roda masuk (entry wheel), rotor, reflektor, di mana bagian yang paling penting adalah rotor, karena rotor merupakan mekanisme utama dalam pengenkripsian yang dilakukan.

Proses enkripsi yang dilakukan Mesin Enigma sebenarnya adalah substitusi, di mana suatu huruf digantikan dengan suatu huruf juga, hanya saja substitusi dilakukan beberapa kali. Meskipun hanya dengan substitusi, suatu pesan akan sulit sekali didekripsi jika tidak dengan alat yang sama, dengan pengaturan posisi yang sama, jenis substitusi yang sama, dan kode kunci yang sama dan semua substitusi dilakukan dengan sambungan listrik melalui kawat (*wiring*). Rotor untuk Mesin Enigma yang digunakan adalah rotor di sisi kiri (L), sisi tengah (M) dan sisi kanan (R), sedangkan rotor lainnya diberi nama rotor I, rotor II, rotor III, dan seterusnya. Jika rotor kiri dihubungkan dengan rotor tengah, maka ketika rotor kiri dialiri listrik, rotor tengah juga dialiri listrik. Secara sederhana, *wiring* menunjukkan subsitusi saklar yang ditekan,

yang dilakukan dengan cara memasang lampu. Jadi, jika saklar pada rotor kiri ditekan, maka lampu rotor tengah menyala. Kemudian proses tersebut diulang dengan mengganti rotor yang sedang digunakan. Kemudian dilakukan pergeseran pada rotor setiap kali ada saklar yang ditekan. Begitu seterusnya selama pesan diketik, dan dengan adanya reflektor, aliran arus dapat dibalikkan dari rotor kanan ke rotor kiri, yang efeknya meningkatkan substitusi sebanyak 26 kali. Reflektor ini menyebabkan Mesin Enigma tidak perlu mengubah keadaan jika ingin mengenkripsi informasi atau ingin mendekripsikannya. Namun reflektor ini menyebabkan kelemahan pada Mesin Enigma di mana terjadi gerak bolak-balik di mana jika misal huruf M dienkripsi menjadi huruf T, maka huruf T dapat dienkripsi menjadi huruf M pada rotor yang sama. Pada setiap rotor terdapat gerakan memutar (turnover), yaitu posisi di mana suatu rotor mulai bergerak menggeser rotor di sampingnya. Rotor bergerak 1 huruf setiap kali tombol ditekan., dan jika gerakan memutar rotor menunjuk huruf S, maka rotor menggeser rotor huruf M sejauh 1 huruf jika sudah mencapai posisi gerakan memutarnya (posisi di huruf S). Setiap jenis rotor mempunyai gerakan memutar masing-masing.

3.8. Program Aplikasi Enkripsi Dan Dekripsi

Program aplikasi penggunaan enkripsi dan dekripsi adalah program *Pretty Good Privacy* (PGP), dan *secure shell* (SSH). Program PGP digunakan untuk mengenkripsi dan menambahkan digital signature dalam e-mail yang dikirim, sedangkan program SSH digunakan untuk mengenkripsi session telnet ke suatu host.

Pretty Good Privacy (PGP) dikembangkan pertamakali oleh Phil Zimmermann pada akhir tahun 1980 untuk melindungi email dengan metode enkripsi dan pengesahan berupa tanda tangan digital. Saat ini, PGP tidak hanya ditujukan untuk keamanan email tetapi juga untuk keamanan berbagai file dan program pada komputer personal (PC). *Pretty Good Privacy* menggunakan kriptografi kunci simetris dan kriptografi kunci publik. Oleh karena itu, PGP mempunyai dua tingkatan kunci, session key yang merupakan kunci rahasia untuk enkripsi data dan pasangannya berupa kunci publik yang digunakan sebagai tanda tangan digital. Kunci simetris hanya dipakai sekali (one-time) dan dibuat secara otomatis. Program PGP dapat di-download melalui internet pada situs http://www.pgp.org. PGP versi awal menggunakan IDEA sebagai algoritma kunci simetris dan RSA sebagai algoritma kunci publik, sedangkan pada versi mutakhir menggunakan algoritma CAST sebagai algoritma kunci simetris serta algoritma Diffie-Hellman sebagai algoritma kunci publik. PGP memiliki fitur untuk menghapus dokumen secara aman, enkripsi pada hard disk, dan enkripsi pada jaringan.

Secara garis besar PGP memiliki tiga fitur utama, yaitu:

1. Fitur untuk melakukan enkripsi dan menandatangani dokumen.
2. Fitur untuk melakukan dekripsi dan verifikasi tanda tangan.
3. Fitur untuk mengelola kunci PGP yang dimiliki oleh pengguna.

Secure shell (SSH) adalah protokol yang dapat memungkinkan komunikasi secara aman antara dua terminal melalui jaringan telekomunikasi karena menyediakan jalur

yang terenkripsi untuk dua terminal yang terhubung. *Secure shell* (SSH) dirancang sebagai pengganti Telnet dan Protokol Remote Shell lainnya yang tidak aman seperti RSH Berkeley dan Protokol REXEC, yang mengirim informasi, terutama kata sandi dalam bentuk teks.

SSH seringkali dibandingkan dengan protokol SSL/TLS karena keduanya berfungsi untuk mengamankan komunikasi. Namun, perbedaan utama antara SSH dan SSL/TLS adalah pada tujuan penggunaannya. SSH digunakan untuk mengamankan koneksi jaringan dan akses ke sistem, sedangkan SSL/TLS digunakan untuk mengamankan koneksi website dan data pengguna.

SSH umumnya digunakan oleh admin sistem untuk mengelola server dari jarak jauh dan perangkat jaringan untuk menyediakan pengesahan, enkripsi, dan integritas data yang kuat. Dengan SSH, pengguna dapat mentransfer file antar komputer dengan aman. Setiap server yang dibuat pada Virtual Private Server (VPS) dan Virtual Data Center (VDC) yang menggunakan Sistem Operasi Linux serta memiliki IP Publik, sudah terdapat fitur SSH secara otomatis. Enkripsi yang digunakan oleh SSH bertujuan untuk memberikan kerahasiaan dan integritas data melalui jaringan yang tidak aman, seperti Internet.

Fungsi SSH adalah sebagai berikut:

1. Sebagai kombinasi Secure File Transfer Protocol (SFTP), yaitu alternatif yang aman untuk transfer file dibandingkan FTP konvensional dengan RSYNC untuk menyalin dan mentransfer file secara efisien dan aman (hampir sama dengan FTP);
2. Sebagai pengamanan ketika menjelajahi web melalui

koneksi proxy yang dienkripsi dengan client SSH yang mendukung protokol SOCKS;

3. Melakukan remote monitoring dan pengelolaan server melalui satu atau lebih mekanisme pengamanan, SSH memungkinkan pengguna untuk melakukan login jarak jauh ke server atau komputer lain yang memungkinkan pengguna mengakses sistem tanpa perlu berada di depan perangkat secara fisik;
4. Sebagai pembuat terowongan yang aman (tunneling) dalam WAN sehingga memungkinkan pengguna untuk mengakses sumber daya jaringan yang terbatas, seperti website yang diblokir, dari lokasi yang jauh.
5. Melaksanakan port forwarding, yaitu proses mengarahkan lalu lintas jaringan dari satu port ke port lainnya sehingga pengguna dapat mengakses sumber daya jaringan di belakang firewall atau router yang membatasi akses.
6. Melakukan proses pengesahan untuk memverifikasi identitas pengguna yang dapat dilakukan dengan menggunakan username dan password, atau menggunakan kunci publik dan privasi.

Teknologi enkripsi yang digunakan dalam SSH adalah enkripsi simetris. Sesuai namanya, enkripsi ini mencari kesamaan antara kunci yang diterima dengan kunci yang ada. Jadi, ketika terminal klien mengirimkan permintaan terhubung, terminal klien akan menerima public key. Server kemudian menanggapinya dengan salinan kuncinya sendiri. Kedua kunci kemudian akan dibandingkan dan jika keduanya simetris atau sama maka komunikasi akan dilanjutkan untuk bertukar data dengan aman. Berbeda dengan enkripsi simetris, enkripsi asimetris bekerja dengan cara mendekripsi pesan dan bukan

mencari kecocokan seperti enkripsi simetris. Ketika terminal klien membuat koneksi, dua kunci yang berbeda akan dihasilkan dimana hanya kunci dari terminal klien yang dapat mengenkripsi pesan dari kunci server. Kunci asimetris bekerja seperti potongan jigsaw; potongan kunci yang diterima terminal klien hanya dapat membuka potongan kunci yang berada pada server. Jadi, ketika kunci dari terminal klien memiliki perubahan data, hal ini akan membuat kunci dari terminal klien tidak dapat lagi membuka pesan dari server.

Dalam SSH terdapat teknik *hashing*. Secara sederhananya, *hashing* adalah proses mengubah data menjadi suatu string atau untaian angka dan huruf yang panjang dan unik. Untaian angka dan huruf ini nantinya dapat diubah kembali menjadi data asli dengan algoritmanya sendiri. Oleh karena itu, jika pesan dicuri atau diubah di tengah komunikasi, maka pesan yang dicuri atau diubah tersebut tidak dapat diubah kembali menjadi untaian angka dan huruf tersebut. SSH pada umumnya menggunakan port 22 untuk akses ke jaringan internet.

Dalam sistem operasi Linux, SSH diakses dengan memanfaatkan terminal dan aplikasi pendukung seperti OpenSSH. Berbeda dengan Linux & Mac yang mengandalkan terminal untuk mengakses SSH, Windows menggunakan third party software untuk dapat menggunakan fitur dari SSH ini, seperti aplikasi Putty.

Dalam pengelolaan SSH, pengesahan pengguna perlu diatur dengan benar, seperti memastikan kata sandi yang digunakan cukup kuat dan memperbarui secara berkala, serta menerapkan pengesahan dua faktor. Selain itu, pengaturan izin akses diatur dengan cermat, dengan memastikan hanya

pengguna yang memiliki izin yang tepat yang dapat mengakses data dan sistem. Selanjutnya, pengaturan sertifikat digital dapat membantu memastikan keamanan dan pengesahan pengguna dengan memeriksa data kredensial digital pengguna. Dengan mengelola SSH secara efektif, organisasi dapat memastikan bahwa keamanan jaringan pengguna tetap terjaga dan risiko serangan berkurang.

Salah satu aspek penting pengelolaan SSH adalah pengaturan izin akses. Izin akses harus diberikan dengan hati-hati, karena pengguna yang tidak berwenang dapat memanfaatkan celah keamanan untuk merusak jaringan. Pengaturan izin akses berdasarkan pada prinsip pengguna hanya diberikan akses yang dibutuhkan untuk menyelesaikan tugas tertentu. Pengelolaan SSH juga melibatkan pengaturan sertifikat digital. Sertifikat digital digunakan untuk memverifikasi identitas pengguna yang terhubung ke server. Sertifikat digital adalah cara yang aman dan terenkripsi untuk memastikan bahwa pengguna yang terhubung adalah pengguna yang sah dan bahwa data yang dikirimkan melalui jaringan aman dari serangan.

Pengelolaan SSH dapat dilakukan dengan menggunakan beberapa komponen, seperti server SSH, pengelola kunci SSH, dan firewall. Konfigurasi server SSH melibatkan pengaturan pengguna dan izin akses, sertifikat digital, dan pengaturan protokol. Pengelola kunci SSH mengelola kunci SSH untuk mengenkripsi data yang dikirimkan melalui jaringan. Firewall digunakan untuk mengatur lalu lintas jaringan dan mencegah akses tidak sah ke jaringan.

Selain, PGP dan SSH, masih terdapat program-program enkripsi dekripsi lainnya, seperti Cryptol. Cryptol adalah

bahasa pemrograman yang digunakan untuk menspesifikasi algoritma kriptografi sebagai penerapan spesifikasi matematika dalam general purpose language.

Implementasi Cryptol tidak ambigu walaupun dideskripsikan dalam bahasa Inggris dan spesifikasi matematika, jika dibandingkan dengan skema SHA-1. Cryptol juga dapat digunakan untuk menganalisa algoritma kriptografi. Dengan begitu programmer dapat melakukan test vector, membuktikan teorema dan menggunakan perangkat lain untuk memverifikasi ekuivalensi aplikasinya. Selain itu, programmer juga dapat mengimplementasikan Cryptol untuk perangkat keras.

Cryptol versi 2 menggunakan lisensi BSD. Cryptol dirancang untuk mengekspresikan berbagai jenis algoritma kriptografi yang dapat diimplementasikan dalam rangkaian perangkat keras. Dalam jaringan komputer, Cryptol juga dapat mengimplementasikan inti kriptografi, misalnya implementasi protokol SSL

BAB IV
EVALUASI KEAMANAN JARINGAN

4.1. Deteksi Probing

Meskipun suatu jaringan telekomunikasi sudah dirancang memiliki perangkat pengamanan, dalam operasinya, keamanan jaringan telekomunikasi harus selalu dimonitor, karena:

1. Dapat ditemukannya lubang keamanan jaringan telekomunikasi (*security hole*) yang baru.
2. Kesalahan konfigurasi.
3. Penambahan perangkat baru (perangkat keras dan/atau perangkat lunak) yang menyebabkan menurunnya tingkat keamanan atau berubahnya metoda untuk mengoperasikan sistem.

Lubang keamanan jaringan telekomunikasi (*security hole*) dapat terjadi karena beberapa hal; seperti: kesalahan dalam perancangan, kesalahan dalam implementasi, kesalahan konfigurasi, dan kesalahan penggunaan. Administrator jaringan telekomunikasi membutuhkan perangkat bantu otomatis, yang dapat membantu menguji atau mengevaluasi keamanan jaringan telekomunikasi yang dikelola.

Layanan di internet umumnya dilakukan dengan menggunakan protokol TCP atau UDP. Setiap layanan dijalankan dengan menggunakan *port* yang berbeda, misalnya:

1. SMTP, untuk mengirim dan menerima *email* (TCP *port* 25),
2. DNS, untuk domain (UDP dan TCP *port* 53),
3. HTTP, untuk web *server* (TCP *port* 80),

4. POP3, untuk mengambil *email* (TCP *port* 110), dan lain-lain.

Untuk beberapa layanan berbasis TCP/IP, proses *probe* dapat dilakukan dengan menggunakan program *telnet*. Untuk mendeteksi adanya *probing* ke suatu jaringan telekomunikasi dapat dipasang suatu program yang mengamati *entry* dalam berkas *log* dapat diketahui adanya *probing.*

4.2. OS Fingerprinting

OS Fingerprinting merupakan istilah yang umum digunakan untuk menganalisa OS dari suatu sistem jaringan telekomunikasi yang dituju. *OS Fingerprinting* dapat dilakukan dengan melakukan *telnet* ke *server* yang dituju atau menggunakan layanan FTP yang tersedia di *port* 21.

Salah satu cara untuk mengetahui kelemahan suatu sistem jaringan telekomunikasi adalah dengan menyerang diri sendiri dengan paket-paket program penyerang (*attack*) yang dapat diperoleh di internet. Selain program penyerang yang sifatnya agresif melumpuhkan sistem yang dituju, terdapat juga program penyerang yang sifatnya melakukan pencurian atau penyadapan informasi. Untuk penyadapan informasi, biasanya dikenal dengan istilah "*sniffer*".

4.3. Eksplorasi Keamanan Jaringan Telekomunikasi

Tahapan yang biasa dilakukan untuk meretas keamanan jaringan telekomunikasi adalah:

1. *Footprinting*

   Merupakan tahapan untuk mencari rincian informasi terhadap sistem jaringan yang akan dijadikan sasaran penyerangan, meliputi pencarian informasi dengan *search*

engine, whois, dan DNS zone transfer.

2. *Scanning*
   Merupakan tahapan untuk mencari pintu masuk yang paling mungkin.
3. *Enumeration*
   Merupakan tahapan untuk meneliti secara intensif terhadap sasaran, dan menemukan *account user* jaringan telekomunikasi, sumber daya jaringan telekomunikasi (*network resource*), dan menemukan aplikasi mana yang perlindungannya lemah.
4. *Escalating Privilege*
   Merupakan tahapan untuk mendapatkan hak akses utama sebagai administrator jaringan telekomunikasi dengan *password cracking* atau *exploit* sejenis *getadmin*.
5. *Covering Tracks*
   Merupakan tahapan untuk menutup jejak yang meliputi pembersihan jejak memasuki jaringan telekomunikasi (*network log*).
6. *Creating Backdoors*
   Merupakan tahapan untuk membuat pintu belakang pada berbagai sistem untuk memudahkan masuk kembali ke sistem ini dengan cara membentuk pengguna *account* palsu, menjadwalkan *batch job*, mengubah *startup file*, dan menanamkan layanan pengendali jarak jauh.

BAB V
KEAMANAN JARINGAN BERBASIS SERVER

5.1. Pengaturan Akses Layanan

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*". Dalam sistem operasi Linux, *password* pengguna disimpan dalam *file text* yang terletak di /etc/passwd, sedangkan dalam sistem operasi Windows, *password* pengguna terletak di c:\windows\sistem32\config dimana *file password* tersebut telah menggunakan algoritma enkripsi.

Kata kunci (*Password*) digunakan sebagai fasilitas untuk mengatur hak akses pengguna dalam menggunakan perangkat dan fasilitas jaringan telekomunikasi. Kriteria pembuatan *password* yang sudah umum diketahui adalah:

1. Jangan menggunakan nama *login*, nama pertama atau akhir beserta variasinya dan nama pasangan atau anak.
2. Jangan menggunakan informasi lainnya yang mudah didapat tentang diri pribadi seperti nomor telepon, tanggal lahir.
3. Gunakan *password* yang merupakan kombinasi antara huruf kapital dan huruf kecil dan angka.
4. Gunakan special "32 karakter ALT", diketikkan dengan menahan tombol Alt ketika mengetik angka antara 128 and 255 pada tombol angka dengan indikator *Num Lock*

on.

5. Gunakan *password* yang mudah diketikkan, tanpa perlu melihat pada *keyboard*.

Untuk mengatur hak akses layanan jaringan telekomunikasi berbasis sistem *server*, perlu dilakukan, yaitu:

1. Mengatur hak akses terhadap penggunaan perangkat, dimana umumnya, pengguna sistem perangkat dapat dibedakan atas:
   a. Pengguna terbatas (*restricted user*)
      Adalah pengguna dengan akses terbatas yang dapat menggunakan perangkat dan menyimpan dokumen tetapi tidak dapat memasang program dan mengubah setting sistem windows.
   b. Pengguna standar (*standard user*)
      Adalah pengguna umum yang dapat mengubah beberapa *file system*, memasang program yang tidak berpengaruh terhadap *file system* windows.
   c. Pengelola jaringan telekomunikasi (*advanced user management*)
      Adalah pengguna dengan akses penuh yang dapat mengatur kelompok pengguna dan hak akses pengguna. Pengguna ini umumnya bertugas sebagai pengelola jaringan telekomunikasi.
2. Menutup layanan yang tidak digunakan
   Seringkali sistem (perangkat keras dan/atau perangkat lunak) diberikan beberapa layanan yang dijalankan sebagai *default*, misalnya, dalam sistem UNIX, layanan-layanan berikut sering dipasang dari *vendor*nya: *finger*, *telnet*, *ftp*, *smtp*, *pop*, *echo*, dan seterusnya. Layanan tersebut tidak semuanya dibutuhkan. Untuk mengamankan

sistem jaringan telekomunikasi, layanan yang tidak diperlukan di *server* (perangkat) sebaiknya dimatikan.

3. Memasang pelindung (proteksi)
   Untuk lebih meningkatkan keamanan jaringan telekomunikasi sistem informasi, pelindung dapat ditambahkan. Penyaringan dapat berupa penyaringan *email*, informasi, akses, atau bahkan *level packet*. Misalnya: dalam sistem UNIX terdapat paket program "*tcpwrapper*" yang dapat digunakan untuk membatasi akses ke layanan atau aplikasi tertentu. Misalnya, layanan untuk "*telnet*" dapat dibatasi untuk untuk sistem yang memiliki nomor IP tertentu, atau memiliki domain tertentu.
4. Memasang *firewall* untuk melakukan penyaringan secara umum.

5.2. Hak Akses Pengguna

Untuk memperoleh titik temu dari kemudahan akses pengguna dengan tingkat keamanan jaringan telekomunikasi yang tinggi, maka diperlukan suatu kebijakan tentang pengaturan pengguna beserta hak yang dapat diperolehnya (*access right policy*). Dalam hal ini dapat membagi pihak yang dapat menggunakan jaringan telekomunikasi menjadi tiga kategori, yaitu pihak pengguna terbatas (*restricted user*) seperti karyawan biasa sebagai pihak pengguna pertama, lalu pengguna yang memiliki otoritas tertentu, seperti manajer pada bidang-bidang yang khusus, direktur, sebagai pihak pengguna menengah (*medium user*) dan yang ketiga adalah pihak administrator sebagai pihak pengguna penuh (*fully user*).

Tentu saja, hak akses untuk karyawan biasa berbeda

dengan hak akses pada tingkatan manajer apalagi administrator. Tingkatan karyawan mungkin hanya dapat mengakses informasi pribadinya saja, seperti identitas karyawan, absensi, honor dan hal-hal yang bersifat privacy. Pada tingkatan menengah, manajer dapat saja memiliki hak akses jaringan telekomunikasi dan hak peggunaan perangkat yang lebih baik dibandingkan tingkatan karyawan, semisal dapat mengakses informasi informasi karyawan yang dipimpinnya, informasi-informasi yang berkaitan dengan hal-hal yang menjadi tugas dan tanggung jawab pengelolaannya, semisal, manajer keuangan dapat melhat informasi-informasi keuangan suatu perusahaan dalam kurun waktu tertentu, yang tentu saja hak akses ini tidak dibuka untuk manajer personalia. Di lain pihak, manajer keuangan tidak dapat mengakses informasi-informasi personalia, misalnya status absensi karyawan, yang aksesnya hanya dapat dilihat oleh manajer personalia. Di atas itu semua, tentu saja adalah administrator yang mengelola keseluruhan basis data yang ada pada suatu perusahaan, instansi atau institusi. Seorang administrator dapat melihat semua basis data, jejak akses pengguna, dan mengatur (mengubah) konfigurasi serta setting perangkat jaringan telekomunikasi yang dikelolanya secara bertanggungjawab.

Dengan pengaturan hak akses ini, yang kemudian disosialisasikan ke seluruh karyawan, maka diharapkan tingkat kenyamanan karyawan dalam menggunakan jaringan telekomunikasi dapat dipelihara dengan baik, karena semua karyawan sudah mengetahui posisinya ada dimana dan hak akses apa saja yang dapat dipakai. Hal ini tentu saja akan memudahkan pengaturan lebih lanjut berkaitan dengan layanan jaringan telekomunikasi tambahan, semisal *messenger*,

internet, printer dan sebagainya. Pastinya karyawan yang memiliki posisi sebagai operator produksi hanya dapat menggunakan perangkat untuk absensi atau melihat honornya dalam satu bulan, dan tidak diperkenankan untuk menggunakan fasilitas layanan internet (untuk mencegah agar karyawan produksi tidak mengakses internet saja seharian dan melupakan tugasnya untuk merakit barang produksi). Seorang karyawan di bagian keuangan mungkin tidak memerlukan messenger, karena selama menghitung keuangan memerlukan ketelitian yang sangat tinggi (untuk mencegah kemungkinan agar uang yang dihitung jumlahnya menjadi berbeda setelah asyik mengobrol melalui messenger).

Dari pengaturan hak akses beserta layanan yang diberikan pada akhirnya akan menentukan standar layanan jaringan telekomunikasi minimal yang harus diberikan ke pengguna atau karyawan. Mungkin *bandwith* yang diberikan ke manajer keuangan akan lebih sedikit dibandingkan ke manajer pemasaran atau manajer PR. Begitu juga dari standar perangkat yang digunakan, dapat saja berbeda antara divisi IT misalnya dengan divisi produksi, dan lain sebagainya. Adanya standar penggunaan jaringan telekomunikasi beserta perangkatnya yang disesuaikan dengan *access right policy* di atas dapat menyenangkan semua pihak, dan menjamin tingkat keamanan informasi yang tinggi, karena akhirnya semua berpulang pada tingkat kemahiran administrator dalam mengelola dan mengatur jaringan telekomunikasinya, serta mencegah kemungkinan adanya ancaman dari luar, seperti virus, trojan atau spyware dan anacaman dari dalam seperti penyalhgunaan perangkat dan jaringan telekomunikasi oleh pengguna yang tidak memiliki hak untuk mengaksesnya.

5.3. Port Layanan TCP/IP

Dalam protokol TCP/IP, suatu *port* adalah mekanisme yang mengijinkan suatu perangkat untuk mendukung beberapa sesi koneksi dengan perangkat lainnya dan program dalam jaringan telekomunikasi. *Port* dapat mengidentifikasikan aplikasi dan layanan yang menggunakan koneksi dalam jaringan telekomunikasi. *Port* juga mengidentifikasikan suatu proses tertentu dimana *server* dapat memberikan layanan ke *client* atau bagaimana *client* dapat mengakses suatu layanan yang ada dalam *server*. *Port* dapat dikenali dengan angka 16-Bit (dua *byte*) yang disebut dengan *port number* dan diklasifikasikan dengan jenis protokol *transport* apa yang digunakan, yaitu: *port* TCP dan *port* UDP. Karena memiliki angka 16-bit, maka total maksimum jumlah *port* untuk setiap protokol transport yang digunakan adalah 65536 port.

Dilihat dari penomorannya, *port* UDP dan TCP dibagi menjadi tiga jenis, yakni sebagai berikut:

1. *Well-known Port*, yang awalnya berkisar antara 0 hingga 255 tapi kemudian diperlebar untuk mendukung antara 0 hingga 1023. Nomor port yang termasuk ke dalam *Well-known Port* seperti dalam Tabel 5.1 merepresentasikan layanan jaringan yang sama.
2. *Registered Port*, merupakan *port* yang digunakan oleh *vendor* perangkat atau jaringan telekomunikasi yang berbeda untuk mendukung aplikasi dan sistem operasi yang dibuat. Kisaran *Registered Port* berkisar dari 1024 hingga 49151 dan beberapa *port* di antaranya adalah *Dynamically Assigned Port*.
3. *Dynamically Assigned Port* merupakan *port* yang

ditetapkan oleh sistem operasi atau aplikasi yang digunakan untuk melayani request dari pengguna sesuai dengan kebutuhan. *Dynamically Assigned Port* berkisar dari 1024 hingga 65536 dan dapat digunakan atau dilepaskan sesuai kebutuhan.

Tabel 5.1. Well-known Port

Port	Jenis Port	Keyword	Nama Protokol
20	TCP, UDP	FTP	*File Transfer Protocol (default data)*
21	TCP, UDP	FTP	*File Transfer Protocol (control)*
23	TCP, UDP	telnet	*telnet*
25	TCP, UDP	SMTP	*Simple Mail Transfer Protocol*
53	TCP, UDP	DNS	*Domain Name System*
67	TCP, UDP	bootpc	*DHCP/BOOTP Protocol Server*
68	TCP, UDP	bootpc	*DHCP/BOOTP Protocol Server*
69	TCP, UDP	TFTP	*Trivial File Transfer Protocol*
80	TCP, UDP	HTTP	*Hyper Text Transfer Protocol*
110	TCP, UDP	POP3	*Post Office Protocol versi 3*
123	TCP, UDP	NTP	*Network Time Protocol*
220	TCP, UDP	IMAP	*Interactive Mail Access Protocol*

Protokol yang terdapat dalam Tabel 5.1 dijelaskan berikut ini:

1. FTP (*File Transfer Protocol*) adalah suatu protokol internet yang berjalan dalam lapisan aplikasi yang merupakan standar untuk pentransferan berkas (*file*) perangkat antar mesin-mesin dalam suatu jaringan telekomunikasi.
2. SMTP (*Simple Mail Transfer Protocol*) merupakan salah satu protokol yang umum digunakan untuk pengiriman

surat elektronik di internet. Protokol ini digunakan untuk mengirimkan surat elektronik.

3. HTTP (*Hypertext Transfer Protocol*) adalah protokol yang digunakan untuk mentransfer dokumen dalam *World Wide Web (WWW)*. Protokol ini adalah protokol ringan, tidak berstatus dan generik yang dapat digunakan berbagai macam jenis dokumen.
4. POP3 (*Post Office Protocol versi 3*) adalah protokol yang digunakan untuk mengambil surat elektronik (*email*) dari *server email.*
5. IMAP (*Internet Message Access Protocol*) adalah protokol standar untuk mengakses/mengambil *email* dari *server*. IMAP memungkinkan pengguna memilih *email* yang akan diambil, membuat folder di *server*, mencari pesan *email* tertentu, bahkan menghapus *email* yang ada.

Penjelasan nomor *port* dan jenis layanannya adalah sebagai berikut:

1. 20: FTP (data)
   Digunakan sebagai tempat data masuk.
2. 21: FTP (*server* FTP)
   Digunakan oleh server *File Transfer Protocol* ketika seseorang mengakses server FTP, maka c*lient* FTP secara *default* melakukan koneksi ke *port* 21.
3. 22: SSH (*Secure shell*)
   Adalah *port* standar untuk SSH, biasanya diubah oleh pengelola *server* untuk alasan keamanan.
4. 23: *Telnet server*
   Adalah *port untuk* menjalankan *server telnet*.
5. 25: SMTP (*Simple Mail Transfer Protocol*)
   Adalah *port server* mail, atau *port* standar yang digunakan

dalam komunikasi pengiriman *email* antara sesama SMTP *Server*.

6. 37: *Time Service*
   Adalah *port* built-in untuk layanan waktu.
7. 53: DNS, *Domain Name Server*.
   Server menggunakan *port* ini untuk penamaan alamat, dan menjawab pertanyaan yang terkait dengan penerjamahan nama domain ke *IP Address*.
8. 67: (UDP) BOOTP
   Adalah DHCP *port* (*server*). Kebutuhan akan *Dynamic Addressing* dilakukan melalui *port* ini.
9. 68: (UDP) BOOTP
   Adalah DHCP *port* yang digunakan oleh *client*.
10. 69: TFTP (*Trivial File Transfer Protocol*)
11. 79: *Finger port*
    Digunakan untuk memberikan informasi tentang sistem, dan *login* pengguna.
12. 80: *World Wide Web* (WWW) atau *Hyper Text Transfer ProtocolI* (HTTP)
    Adalah *port server* web yang paling umum digunakan di internet.
13. 81: *World Wide Web* (WWW) atau *Hyper Text Transfer ProtocolI* (HTTP) Alternatif
    Digunakan ketika *port* 80 diblok maka *port* 81 dapat digunakan sebagai *port* altenatif untuk melayani HTTP.
14. 98: *port* administrasi akses web Linuxconf port.
15. 110: POP3 (*Post Office Protocol*)
    Adalah *port server* pop mail ketika mengambil *email* yang tersimpan di *server* dengan menggunakan teknologi POP3 yang berjalan di *port* ini.

16. 111: sunrpc (*Sun Remote Procedure Call*) atau *mapper port*
    Digunakan oleh NFS (*Network File System*), NIS (*Network Information Service*), dan berbagai layanan terkait.
17. 113: identd atau *auth port server*
    Kadang-kadang diperlukan oleh beberapa layanan bentuk lama (seperti SMTP dan IRC) untuk melakukan validasi koneksi.
18. 119: NNTP atau *port* yang digunakan oleh News *Server*, sudah sangat jarang digunakan.
19. 123: *Network Time Protocol* (NTP)
    Adalah *port* yang digunakan untuk sinkronisasi dengan *server* waktu dimana tingkat akurasi yang tinggi diperlukan.
20. 137-139: *NetBIOS* (SMB).
21. 143: IMAP, *Interactive Mail Access Protocol*
    Merupakan aplikasi yang memungkinkan untuk membaca *email* yang berada di *server* dari perangkat di rumah/kantor, protokol ini sedikit berbeda dengan POP.
22. 161: SNMP, *Simple Network Management Protocol*
    Lebih umum digunakan di *router* dan switch untuk memantau statistik dan tanda-tanda vital (keperluan pemantauan).
23. 177: XDMCP, *X Display Management Control Protocol*
    Digunakan untuk sambungan *remote* ke suatu X *server*.
24. 443: HTTPS, HTTP (WWW) yang aman dan cukup lebar.
25. 465: SMTP atas SSL, protokol *server email*
26. 512 (TCP): exec
    Adalah bagaimana tampilan di netstat. Sebenarnya nama yang tepat adalah rexec, untuk *Remote Execution*.

27. 512 (UDP): biff
    Adalah protokol untuk *email* pemberitahuan.
28. 513: *Login*, sebenarnya *rlogin*, alias *Remote Login*
    Tidak ada hubungannya dengan *standar/bin/login* yang digunakan setiap kali *login*.
29. 514 (TCP): *Shell*
    Adalah nama panggilan, dan bagaimana netstat menunjukkan hal itu. Sebenarnya, rsh adalah aplikasi untuk "*Remote* Shell". Seperti semua "r" perintah ini melemparkan kembali ke kindler, sangat halus.
30. 514 (UDP): *Daemon syslog port*
    Hanya digunakan untuk tujuan *logging remote*.
31. 515: lp atau mencetak *port server*.
32. 587: MSA, *Mail Submission Agent*
    Adalah suatu protokol penanganan surat baru didukung oleh sebagian besar MTA's (*Mail Transfer Agent*).
33. 631: CUPS (Daemon untuk keperluan printing)
    Adalah *port* yang melayani pengelolaan layanan berbasis web.
34. 635: Mountd, bagian dari NFS.
35. 901: SWAT, Samba Web Administration Tool port
    Adalah *port* yang digunakan oleh aplikasi pengelolaan SAMBA berbasis web.
36. 993: IMAP melalui SSL.
37. 995: POP melalui SSL.
38. 1024: *port* pertama yang merupakan *Unprivileged port*
    Ditugaskan secara dinamis oleh kernel untuk aplikasi apapun yang memintanya. Aplikasi lain umumnya menggunakan *port unprivileged* di atas *port* 1024.
39. 1080: *Socks Proxy Server*.

40. 1433: MS SQL *port server*.
41. 2049: NFSd, *Network File Service Daemon port*.
42. 2082: *port* cPanel
    Port ini digunakan untuk aplikasi pengelolaan berbasis web yang disediakan oleh cpanel.
43. 2095: *port* ini di gunakan untuk aplikasi webmail cpanel.
44. 2086: *port* ini di gunakan untuk WHM, atau *Web Host Manager cpanel*.
45. 3128: *port server Proxy Squid*.
46. 3306: *port server* MySQL.
47. 5432: *port server* PostgreSQL.
48. 6000: X11 TCP *port* untuk *remote*.
    Mencakup *port* 6000-6009 karena X dapat mendukung berbagai menampilkan dan setiap tampilan akan memiliki *port* sendiri. SSH X11 *Forwarding* akan mulai menggunakan *port* pada 6.010.
49. 6346: Gnutella.
50. 6667: ircd, *Internet Relay Chat Daemon*.
51. 6699: Napster.
52. 7100-7101: Beberapa *font server* menggunakan *port* tersebut.
53. 8000 dan 8080: *Common Web Cache* dan *port server Proxy* Web.
54. 10000: Webmin, *port* yang digunakan oleh webmin dalam layanan pengelolaan berbasis web.

5.4. Keamanan Mail Server

Email sebagai salah satu layanan dalam jaringan telekomunikasi sudah digunakan orang sejak awal terbentuknya internet di tahun 1969. Alamat *email* merupakan

gabungan dari nama pengguna dan *domain name*: user@*domainname.* Misalnya: sri@gmail.com.

Mail server hanya suatu aplikasi yang berurusan dengan lalu lintas *email*, tidak secara langsung berhubungan dengan pengguna yang akan berkirim *email*.

Sistem *email* memiliki dua komponen, yaitu:

1. Agen Pengguna Surat/*Mail User Agent (MUA)*
   Berhubungan dengan pengguna, misalnya: *Pine, Eudora, Netscape, Outlook dan Pegasu*s.
2. Agen Pentransfer Surat/*Mail Transfer Agent (MTA)*
   Berhubungan dengan pengiriman *email*, misalnya: *sendmail, qmail, Exim, postfix, Mailer daemon, exchange*.

Email terdiri dari tiga komponen, yaitu:

1. *Envelope,* atau amplop. Ini digunakan oleh MTA untuk pengiriman. Envelope ditandai dengan dua perintah SMTP:
   MAIL from: sri@gmail.com RCPT to: tanto@gmail.com.
2. *Header,* digunakan oleh *user agent*. terdapat kurang lebih sembilan *field header*, yaitu: *Received, Message-Id, From, Date, Reply-To, X-Phone, X-mailer, To* dan *Subject*. Setiap *field header* berisi suatu nama yang diikuti oleh suatu titik dua (:), dan nilai dari field header tersebut.
3. *Body* merupakan isi informasi dari pengirim ke penerima.

Ancaman keamanan terhadap *mail server* adalah:

1. Penyadapan
   Penyadapan dapat terjadi di setiap titik jaringan telekomunikasi yang dilalui. Perlindungan terhadap penyadapan adalah dengan menggunakan enkripsi untuk mengacak isi surat. Contoh perlindungan: *Pretty Good Privacy (PGP*), PEM.

2. *Email* Palsu

   Mudah membuat *email* palsu dengan membuat header sembarang. *Email* palsu ini kemudian dikirimkan via MTA atau langsung via SMTP. Kegiatan tercatat di *server* dalam berkas *log*. Perlindungan terhadap *email* palsu adalah dengan:

   a. Lihat *header* untuk mengetahui asal *email,*
   b. Menggunakan *digital signature*,
   c. Namun keduanya jarang dilakukan.

3. *Mailbomb*

   Dengan mengirim banyak *email* ke satu alamat *email*. Perlindungan terhadap *mailbomb* adalah dengan membatasi ukuran *email*, quota disk, menggunakan penyaringan khusus.

4. *Mail Relay*

   Menggunakan *server* orang lain untuk mengirimkan *email* yang mengakibatkan *bandwidth* jaringan telekomunikasi terpakai untuk mengirim banyak *email*.

BAB VI
ANCAMAN TERHADAP KEAMANAN JARINGAN

6.1. Pengelolaan Resiko

Pengelolaan terhadap keamanan jaringan telekomunikasi dapat dilihat dari sisi pengelolaan resiko (*risk management*). Terdapat tiga komponen yang memberikan kontribusi terhadap resiko kemanan jaringan, yaitu: kepengelolaan (*asset,*) kelemahan/kerentanan yang ditemui dalam suatu jaringan (*vulnerabilities*) dan ancaman keamanan yang berasal dari luar (*threats*).

Terdapat beragam permasalahan dalam jaringan telekomunikasi yang perlu diperhatikan dan ditangani, di antaranya adalah yang berkaitan dengan keamanan informasi atau informasi yang dikirimkan melalui jaringan telekomunikasi, serta kehandalan suatu jaringan lokal memberikan layanan bagi pengguna atau pelanggan agar dapat berkomunikasi dengan baik. Dalam mempelajari keamanan dan kehandalan pada suatu jaringan telekomunikasi, terdapat enam aspek utama yang perlu diperhatikan dalam pengelolaan suatu jaringan telekomunikasi.

6.2. Kategori Ancaman Terhadap Jaringan

Ancaman terhadap keamanan jaringan telekomunikasi, yang diakibatkan oleh adanya beragam ancaman (*threats*) dan kelemahan (*vulnerabilities*) dari sistem jaringan telekomunikasi, di antaranya adalah:

1. Ancaman keamanan yang dapat menyebabkan terjadinya penolakan akses ke beberapa layanan atau sumber yang diberikan oleh suatu sistem.
2. Ancaman keamanan yang memungkinkan penyusup (*intruder)* dapat mengoperasikan suatu sistem dengan hak yang tidak sah (*unauthorized previlleges*).
3. Percobaan memasuki suatu sistem jaringan telekomunikasi untuk mencari kelemahan potensial yang terdapat dalam sistem jaringan telekomunikasi.
4. Ancaman keamanan secara fisik yang perlu dihadapi oleh perangkat komunikasi atau perangkat.
5. Ancaman keamanan yang disebabkan oleh *worm* atau *virus*.

Tantangan atau ancaman (*threats*) yang dihadapi berkaitan dengan jaminan akan ketersediaan informasi *(ensure information availability*) dapat berupa ancaman keamanan penolakan layanan atau *Denial of Service (*DoS*) attack*. Terdapat beragam jenis ancaman keamanan DoS, tetapi dapat dikelompokkan ke dalam tiga jenis yang utama, yaitu ancaman keamanan penolakan layanan secara umum atau *Denial of Service (*DoS*)*, ancaman keamanan penolakan layanan tersebar atau *Distributed Denial of Service (D*DoS*)*, dan ancaman keamanan penolakan layanan dalam bentuk denyut (berlanjut) atau *Pulse Denial of Service (P*DoS*)*.

Terdapat beberapa kemungkinan ancaman keamanan (*attack*), yaitu:

1. *Interruption*, perangkat sistem menjadi rusak atau tidak tersedia.
2. *Interception*, pihak yang tidak berwenang berhasil mengakses aset atau informasi.

3. *Modification*, pihak yang tidak berwenang tidak saja berhasil mengakses, akan tetapi dapat juga mengubah (tamper) aset.
4. *Fabrication*, pihak yang tidak berwenang menyisipkan objek palsu ke dalam sistem.

6.3. Serangan Penolakan Layanan

Salah satu ancaman keamanan jaringan telekomunikasi adalah serangan penolakan layanan atau *Denial of Service* (DoS) yang merupakan salah satu bentuk ancaman (*threats*) dalam jaringan telekomunikasi. DoS merupakan suatu usaha untuk melumpuhkan sistem jaringan telekomunikasi yang dijadikan target ancaman keamanan sehingga sistem jaringan telekomunikasi tidak dapat menyediakan layanan-layanannya, atau dapat saja tingkat kualitas layanan menjadi menurun dengan drastis dalam waktu yang singkat.

Ancaman keamanan tradisional DoS merupakan ancaman keamanan yang membanjiri (*flooding based* DoS) suatu *server* dengan paket-paket informasi tidak bermanfaat (*useless packets)* dalam jumlah yang tidak dapat dikendalikan. Lebih lanjut, beberapa ancaman keamanan DoS berikutnya dengan kecepatan lebih rendah dikirimkan. Ancaman keamanan-ancaman keamanan baru ini dapat menyerang aliran TCP lebih efektif bila dibandingkan ancaman keamanan sebelumnya.

Ancaman keamanan DoS (DoS *Attack)* merupakan suatu penggunaan layanan yang tidak semestinya oleh seseorang atau sekelompok orang pada suatu layanan yang dihubungkan ke internet secara aktif (*online*) oleh suatu penyedia layanan (*server*) dengan tujuan untuk mengganggu pihak lain yang menikmati layanan yang disediakan oleh *server* tersebut.

Ancaman keamanan dalam bentuk DoS dapat dilancarkan ke semua layanan, seperti suatu *web server* yang memberikan halaman layanan di internet, dan jaringan telekomunikasi, seperti suatu jaringan telekomunikasi yang dihubungkan ke suatu *server*. Akibat dari ancaman keamanan DoS dapat bermacam-macam kemungkinannya, dari hal yang terkecil, seperti ketidaknyamanan (*incovenience*) yang diterima oleh seorang pengunjung (pengguna) suatu halaman layanan di internet *(website),* hingga hal yang serius, seperti kerugian dalam keuangan (*financial losses*) suatu perusahaan yang dipercayakan pada ketersediaan layanan informasi keuangan secara *online* dalam suatu *website* untuk melakukan suatu transaksi bisnis. Ancaman keamanan DoS dapat dibedakan ke dalam beberapa jenis, yaitu *TCP SYN Flood Attack*, *UDP Flood Attack*, *Ping Of Death Attack*, *Smurf Attack*, *Teardrop Attack*, *Bonk Attack*, dan *Land And Latierra Attack*.

Ancaman keamanan DoS dapat dicegah dengan menggunakan berbagai macam mekanisme atau metode penanganan sebelum terjadinya ancaman keamanan (*preventing of attack*). Inti dari mekanisme pencegahan tersebut sebenarnya adalah menutup atau memperbaiki kelemahan suatu sistem yang kemungkinan dapat dimanfaatkan untuk melancarkan ancaman keamanan, dalam hal ini ancaman keamanan penolakan layanan (*Denial of Service*).

Denial of Service adalah serangan yang bertujuan untuk menghambat kerja suatu layanan (servis) atau mematikannya, sehingga pengguna yang berhak mendapatkan layanan menjadi tidak dapat menggunakan layanan tersebut.

Pada dasarnya, untuk melumpuhkan suatu layanan dibutuhkan penggunaan sumber daya (*resource*) yang besar,

sehingga perangkat/mesin yang diserang kehabisan *resource* dan menjadi hang. Beberapa jenis *resource* yang dihabiskan diantaranya adalah:

1. *Swap Space*
   Hampir semua sistem menggunakan ratusan MBps *Swap Space* untuk melayani permintaan *client*. Bagaimanapun *Swap Space* selalu berubah dan digunakan dengan sangat berat. Beberapa serangan *Denial of Service* mencoba untuk memenuhi (mengisi) *Swap Space* ini.
2. *Bandwidth*
   Beberapa serangan *Denial of Service* menghabiskan *bandwidth*.
3. *Kernel Tables*
   Serangan pada *kernel tables*, dapat berakibat sangat buruk pada sistem.
4. Alokasi memori ke kernel juga merupakan target serangan yang sensitif. Kernel memiliki kernelmap limit, jika sistem mencapai posisi ini, maka sistem tidak dapat lagi mengalokasikan memory untuk kernel dan sistem harus di *reboot*.
5. RAM
   Serangan *Denial of Service* banyak menghabiskan RAM sehingga sistem mau-tidak mau harus di re-boot.
6. Disk
   Serangan klasik banyak dilakukan dengan memenuhi Disk.
7. INETD
   Sekali saja INETD rusak, semua *service* (layanan) yang melalui INETD tidak akan bekerja.

Ancaman keamanan *TCP SYN Flood Attack* atau yang terkadang disebut dengan *TCP Half Open* dapat dilakukan

dengan mengandalkan kelemahan atau efek samping dalam metode peralihan informasi antara perangkat *server* dengan perangkat *client* melalui jaringan internet, yang dikenal dengan metode atau mekanisme *three way handshake*. Dalam ancaman keamanan *TCP SYN Flood Attack*, suatu perangkat *server* target dibanjiri dengan tanda SYN (*synchronization*), yang merupakan bagian pertama dari mekanisme *three way handshake*, dalam proses yang mengawali suatu hubungan menggunakan *Transmission Control Protocol (TCP)*.

Dalam proses peralihan informasi yang sebenarnya atau yang umum terjadi, *three way handshake* dimulai dengan pengiriman paket informasi yang ditandai dengan *header SYN* oleh perangkat penerima layanan (perangkat *client)* yang hendak mengirimkan informasi ke perangkat *server*. Dalam proses kedua, perangkat *server* menjawab permintaan perangkat *client* dengan mengirimkan kembali paket informasi yang dikirimkan oleh perangkat *client* tadi dengan dibubuhkan tanda SYN dan ACK *(acknowledgement).* Dalam proses yang terakhir, perangkat *client* mengirimkan kembali tanda ACK yang dikirimkan oleh perangkat *server*, sebagai deklarasi bahwa hubungan atau hubungan antara perangkat *client* ke perangkat *server* dalam suatu jaringan telekomunikasi diperbolehkan dan telah terbuka. Hubungan akan terus berjalan hingga salah satu pihak mengirimkan paket informasi dengan tanda *FIN (finish)* atau *RST (reset)* yang menyatakan bahwa hubungan antara dua perangkat dalam jaringan telekomunikasi telah tertutup *(connection time out)*. Selama belum ada pengiriman paket informasi yang ditandai dengan SYN atau RST, pertukaran informasi antara dua perangkat masih akan terus terjadi, dimana selama pertukaran informasi tersebut,

masih terdapat tanda-tanda lain yang dikirimkan sebagai parameter yang dibutuhkan agar hubungan dapat berjalan dengan baik.

Kelemahan *three way handshake* ini kemudian dimanfaatkan dalam *TCP SYN Flood Attack*, dengan pengiriman sebanyak mungkin paket mengawali hubungan (paket yang ditandai dengan SYN untuk memulai paket informasi) ke perangkat *server* target secara terus menerus. Akibatnya *host,* dalam hal ini perangkat *server* target mengalokasikan memori yang digunakan untuk menerima paket mengawali hubungan tersebut. Dikarenakan paket mengawali hubungan ini dikirimkan terus menerus oleh perangkat *client* penyerang, maka kapasitas ruang memori perangkat *server* target yang disediakan untuk menerima paket mengawali hubungan habis atau penuh. Hal ini tentu saja dapat menyebabkan tidak adanya ruang memori yang tersisa untuk melayani permintaan melakukan hubungan berikutnya, meskipun permintan baru tersebut adalah permintaan layanan yang sebanarnya (bukan lanjutan ancaman keamanan *TCP SYN Flood Attack*). Dengan kata lain, perangkat *server* tidak sanggup atau tidak dapat lagi melayani permintaan hubungan yang baru, sehingga terjadi penolakan layanan (*Denial of Service*) yang dilakukan oleh perangkat *server*.

Untuk menghadapi ancaman keamanan penolakan layanan (*Denial of Service*) dalam bentuk *TCP SYN Flood Attack*, beberapa sistem operasi dapat mengantisipasinya dengan:

1. *Micro Block*

   Ketika ada suatu *host* menerima paket mengawali hubungan, maka *host* akan mengalokasikan ruang memori

yang sangat kecil, sehingga *host* tersebut dapat menerima hubungan lebih banyak. Diharapkan ruang memori dapat menampung semua hubungan yang dikirimkan, sampai terjadi *connection time out*, dimana hubungan-hubungan yang tidak dapat menyelesaikan proses *three way handshake* secara keseluruhan, atau tidak melakukan transaksi informasi dalam kurun waktu yang telah ditentukan, padahal hubungan sudah terbuka, maka hubungan tersebut akan diputus, dan paket dengan tanda *SYN* kemudian dihapuskan dari ruang memori, agar dapat memberikan ruang bagi permintaan hubungan yang baru. Metode ini tidak terlalu efektif, karena bergantung pada kecepatan ancaman keamanan yang dilakukan. Apabila ancaman keamanan paket mengawali hubungan dikirimkan lebih cepat dari lamanya waktu yang diperlukan untuk menunggu pemutusan hubungan, maka pada akhirnya ruang memori yang dialokasikan akan habis juga.

2. *SYN Cookies*

   Ketika menerima paket mengawali hubungan, *host* penerima akan mengirimkan paket permintaan konfirmasi yang harus dijawab oleh pengirim, sebelum *host* mengalokasikan memori yang dibutuhkan. Konfirmasi yang diminta berupa paket *SYN-ACK* dengan nomor urut khusus yang merupakan hasil fungsi *hash* dengan masukan berupa alamat IP pengirim, nomor *port* dan lain-lain. Jawaban dari pengirim paket mengawali hubungan harus berisikan informasi-informasi tersebut. Untuk melakukan perhitungan *hash* memerlukan sumber daya komputasi yang cukup besar, sehingga banyak *server* yang

aplikasinya membutuhkan kemampuaan komputasi yang tinggi, tidak dapat menggunakan metode ini. Metode ini mengubah waktu pengalokasian memori, yang seharusnya merupakan proses awal menjadi proses akhir dalam *three way handshake*. Diperlukan cara yang lebih baik untuk menentukan urutan paket tersebut, sehingga sulit untuk ditebak. Jadi, kemungkinannya, metode ini dapat digunakan pada seluruh perangkat jaringan telekomunikasi atau sistem operasi.

3. *RST Cookies*

   Sebagaimana halnya dengan metode *SYN Cookies*, *RST Cookies* juga mengirimkan paket permintaan konfirmasi yang yang harus dijawab oleh pengirim, sebelum *host* mengalokasikan memori yang dibutuhkan. Hanya saja, paket tersebut adalah paket yang salah. Apabila pengirim paket mengawali hubungan adalah pengirim yang sah, pengirim akan mengirimkan paket RST lalu mengulang kembali hubungan. Ketika *host* menerima paket RST, *host* tersebut mengetahui bahwa pengirim tersebut adalah pengirm paket yang sah, dan akan menerima hubungan dari pengirim secara normal. Kelemahan dalam metode ini adalah ketidaksesuaian penggunaan sistem operasi yang berbeda-beda dalam jaringan internet. Selain dengan antisipasi yang dilakukan oleh sistem operasi, ancaman keamanan *TCP SYN Flood Attack* juga dapat dicegah dengan meng-install *Ingress* atau *Egress Router Fiter* untuk menghindari beberapa *IP Spoofing* lokal.

Pada *User Datagram Protocol (UDP)* terdapat dua layanan, yaitu *echo*, yang didefinisikan sebagai karakter-karakter yang telah diterima dan dikembalikan ke pengirim,

serta *chargen*, yang didefinisikan sebagai karakter-karakter yang dihasilkan. Kedua layanan UDP tersebut digunakan di akhir pengetesan kondisi jaringan telekomunikasi, yang akan digunakan untuk melewatkan informasi antara kedua perangkat jaringan telekomunikasi, dimana hal ini dimungkinkan (*enabled*) secara pasti (*default*) pada hampir semua sistem jaringan telekomunikasi. Meskipun demikian, kedua layanan ini dapat dimanfaatkan untuk melancarkan suatu ancaman keamanan DoS dengan menghubungkan atau menghubungkan *port* yang digunakan untuk *chargen* ke *port* untuk *echo* dalam suatu perangkat atau perangkat komunikasi yang sama atau berbeda, sehingga menghasilkan sejumlah besar lalulintas informasi yang melewati jaringan telekomunikasi (*large amount of network traffic*).

Penyerang yang melancarkan ancaman keamanan *UDP Flood Attack* dapat memanfaatkan kelemahan sistem operasi Windows, yaitu dengan mengirimkan sebagian bingkai *IP (IP Fragments)* sasaran (target ancaman keamanan) yang dibentuk salah (*malformed*) yang kemudian dirakit kembali ke dalam Datagram UDP yang tidak tepat *(invalid UDP Datagram*). Pada saat sasaran menerima *invalid UDP Datagram*, layar perangkat yang dimiliki sasaran, dalam hal ini *server* target akan berwarna biru, yang dikenal dengan *bluescreen* atau *freezes*, dimana istilah ini menyatakan kondisi *hang* pada perangkat atau perangkat komunikasi. Dalam kondisi seperti ini, *server* target tidak dapat melayani permintaan layanan yang baru, dan akan menolak permintaan layanan tersebut (*Denial of Service*).

Untuk mencegah kemungkinan munculnya ancaman keamanan *UDP Flood Attack*, yang pertama harus dilakukan

adalah dengan menghilangkan (*disable*) kedua layanan UDP. Kedua layanan UDP tersebut adalah berupa *echo* dan *chargen* yang digunakan untuk pengetesan akhir kondisi jaringan telekomunikasi, dengan mengetikkan perintah/*etc/inetd.conf* di *Console* dalam sistem operasi Linux dan *no udp small services* pada perangkat *Cisco IOS Router*. Kemudian, lalulintas paket informasi yang melalui protokol transpor UDP disaring (penyaringan*ed*) dengan tingkatan *firewall* tertentu. Hanya lalulintas yang sah *(legitimate)* saja yang diperbolehkan melewati *port* 53, yang memberikan layanan *Doman Name System (DNS)*, yaitu layanan penamaan suatu web *server* berdasarkan *IP Public* yang digunakan.

Pembagian suatu paket informasi menjadi beberapa paket informasi berjumlah kecil, atau yang dikenal dengan istilah *fragmentation*, diperlukan pada saat besarnya *Datagram IP* melebihi besarnya unit transmisi maksimum atau *Maximum Transmission Unit (MTU)*. MTU ini terdapat pada suatu segmentasi jaringan telekomunikasi yang memerlukan adanya suatu informasigaram IP. Untuk mensukseskan pera n kembali paket-paket informasi di akhir penerima, suatu *offset* diberikan pada *header* IP untuk setiap *fragment* yang nantinya digunakan untuk mengidentifikasi urutan posisi *fragment* dalam paket informasi keseluruhan.

Teardrop attack memanfaatkan jalur *Internet Protocol (IP)* yang membutuhkan suatu paket dalam jumlah yang besar untuk *router* berikutnya. Jalur IP ini digunakan untuk menangani pembagian beberapa bingkai paket informasi (*fragments)*. Bingkai paket informasi ini akan mengidentifikasi suatu *offset* dalam permulaan paket, yang menyatakan urutan dari paket-paket informasi yang dikirimkan, dan

memungkinkan paket-paket informasi yang masuk dalam sistem penerima dapat dirakit kembali sesuai dengan urutannya.

Penyerang IP memilih suatu nilai *offset* yang membingungkan (c*onfusing offset value*) dalam *fragment* yang kedua atau *fragment* yang terakhir. Dalam suatu ancaman keamanan *teardrop*, bingkai paket dibuat hingga menghasilkan *overlapping offset* yang menyebabkan *host* menjadi *hang* atau *crash* pada saat mencoba untuk merakitnya kembali. Dengan kata lain, ancaman keamanan *teardrop* mengacak-acak nilai *offset*, dan mengakibatkan urutan paket-paket informasi tidak dapat dirakit kembali secara teratur, atau sesuai dengan susunan aslinya, saat dikirimkan oleh pengirim. Andaikan pada akhirnya, *offset-offset* tersebut menemukan kembali nomor urutan yang sebenarnya, namun hal ini telah menghabiskan sumber daya yang ada. Selama perangkat *server* sasaran atau target ancaman keamanan mencoba menemukan kembali *offset* yang sebenarnya, sistem jaringan telekomunikasi yang dilayani oleh *server* tersebut tidak dapat melayani adanya paket-paket informasi berikutnya yang masuk, meskipun *fragmenetation* paket informasi yang masuk terakhir tersebut tidak diacak *offset*-nya. Jika sistem operasi tidak memiliki perencanaan untuk menghadapi situasi tersebut, hal ini dapat membuat sistem menjadi rusak (*crash*), dan berakibat pada penolakan layanan (*Denial of Service*).

Bonk Attack merupakan ancaman keamanan *teardrop* yang dikhususkan dalam sistem operasi *Windows*. Ancaman keamanan tersebut merusak paket-paket informasi UDP yang melewati *port* 53, yang khusus diperuntukkan untuk layanan *Domain Name Sistem (DNS)*. Ancaman keamanan *Bonk Attack*

dapat menyebabkan sistem jaringan telekomunikasi sasaran tidak dapat menentukan urutan paket-paket informasi yang diterima, sehingga akan menimbulkan *crash.* Apabila sistem jaringan telekomunikasi telah *crash*, maka akan menyebabkan terjadinya penolakan layanan (*Denial of Service*).

Ping of Death Attack adalah suatu bentuk ancaman keamanan yang mengeksploitasi program *Packet Internet Gropher (PING)* dengan mengirimkan paket-paket informasi PING dalam jumlah yang banyak ke sistem jaringan telekomunikasi yang dituju. Program PING umumnya terdapat pada berbagai sistem operasi, baik itu *Windows* maupun *Linux*. Spesifikasi pada protokol yang digunakan di jaringan internet, dalam hal ini *Transmission Control Protocol/Internet protocol (TCP/IP)* dapat melewatkan paket informasi dengan ukuran maksimum 65.536 *octet* (satu *octet* setara dengan delapan bit).

Ping of Death Attack dapat dilakukan dengan mengirimkan paket-paket PING melebihi ukuran *Datagram* *net Control Message Protocol (ICMP)*, yang dibungkus dalam paket IP ke sasaran sasaran. Beberapa sistem akan sibuk (*crash*, *freeze*, *reboot*) selama menerima paket informasi yang melebihi ukuran yang sewajarnya ini, dan akan menghasilkan penolakan layanan (*Denial of Service*).

Smurf Attack merupakan ancaman keamanan yang menggunakan permintaan *Packet Internet Gropher (PING Request)* ke semua alamat yang ada dalam suatu jaringan telekomunikasi secara *broadcast.* Seluruh perangkat (*device*) yang berada dalam alamat yang di-*broadcast* tersebut akan menjawab permintaan PING tersebut. Jika suatu sistem jaringan telekomunikasi memiliki banyak perangkat (*device*) dan PING yang di-*broadcast* ini dilakukan terus menerus,

sistem jaringan telekomunikasi dapat dipenuhi oleh tanggapan-tanggapan dari permintaan PING tersebut. Hal ini akan mengakibatkan *bandwith* yang dimiliki jaringan telekomunikasi akan berkurang atau bahkan habis, sehingga jaringan telekomunikasi menjadi lambat atau bahkan *hang.*

Smurf attack biasanya dilakukan dengan menggunakan *IP Spoofing* dan untuk mencegahnya, *router* yang ada perlu dikonfigurasi untuk menolak (*deny*) lalulintas informasi yang mem-*broadcast* (menyebarkan) alamat IP dalam sistem jaringan telekomunikasi yang dikelola, dimana kemungkinannya ancaman keamanan berasal dari jaringan luar. Hampir pada semua kasus, fungsi dari IP yang di-*broadcast* tidak dibutuhkan. Kemudian, *host* atau perangkat *server* perlu dikonfigurasi melalui *variable kernel* untuk tidak mengulangi (*not reply*) paket informasi ke pengirim paket yang menyebarkan alamat IP. Pencegahan yang terakhir adalah dengan mengkonfigurasi *Ingress* atau *Engress* Penyaringan pada *Router* yang dapat mengantisipasi adanya *IP Spoofing*.

Land attack merupakan ancaman keamanan ke sistem jaringan telekomunikasi target dengan menggunakan program yang bernama *Land*. Program *Land* menyerang *server* yang dituju dengan mengirimkan paket paket informasi palsu yang seolah-olah berasal dari *server* yang akan dijadikan sasaran. Dengan kata lain, sumber dan tujuan dari paket informasi seolah-olah berasal dari *server* yang menjadi sasaran (seolah-olah paket informasi hanya berputar-putar saja dalam perangkat *server* sasaran). Akibatnya, *server* yang diserang akan bingung (*confuse*).

Perbaikan *Land Attack* adalah dengan mengganti program *Land* dengan program *Latierra* untuk melancarkan

ancaman keamanan (mengirimkan paket-paket informasi palsu dalam jumlah yang tidak dapat dikendalikan) ke perangkat target. Dalam program Latierra, nomor *port* yang digunakan dapat berubah-ubah sesuai dengan tujuan ancaman keamanan, sehingga menyulitkan dalam antisipasi pengamanannya, karena tidak dapat diketahuinya secara pasti *port* mana yang akan ditutup untuk menghindari masuknya atau membanjirnya paket paket informasi ancaman keamanan ke dalam perangkat *server* target.

6.4. Serangan Penolakan Layanan Tersebar

Ancaman serangan penolakan layanan tersebar/*Distributed Denial of Service (DDoS)* merupakan ancaman keamanan DoS dalam skala yang lebih besar, dimana ancaman keamanan dilaksanakan secara terkoordinasi menggunakan ketersediaan layanan yang terdapat dalam sistem jaringan telekomunikasi sasaran.

Ancaman keamanan DDoS dilancarkan dengan pengiriman suatu paket dengan *volume* yang sangat besar secara ekstrim ke mesin target melalui kerjasama secara simultan (*simultaneous cooperation*) sejumlah *host* yang tersebar pada jaringan internet. Lalu lintas ancaman keamanan yang begitu besar akan menghabiskan sumber daya *bandwidth* jaringan telekomunikasi atau sumber daya *computing* pada *host* target ancaman keamanan, sehingga permintaan hubungan yang sebenarnya atau yang sah tidak dapat dilayani *(discarded)*. Akibat dari ancaman keamanan ini dapat menyebabkan ketidaknyamanan pengunjung suatu *website* dalam skala yang kecil, hingga dalam skala yang lebih besar berupa kehilangan informasi keuangan yang dipercayakan pada

layanan *online* melalui jaringan internet.

Ancaman keamanan *Distributed Denial of Service (D*DoS*)* muncul sebagai suatu cara yang lazim digunakan untuk mematikan kegiatan suatu organisasi di internet dan menghasilkan kerugian keuangan *(financial losses*) pada waktu yang bersamaan. Dalam ancaman keamanan DDoS, seorang musuh mencoba untuk memutuskan hubungan salah satu elemen (perangkat) dalam jaringan telekomunikasi dengan memetuskan sambungan *(link)* atau titik simpul *(node)* jaringan telekomunikasi.

Ancaman keamanan *Distributed Denial of Service (D*DoS*)* adalah ancaman keamanan yang dikirimkan dari berbagai sistem sumber (*multiple source sistem*). Jika penyerang dapat mengelola sejumlah besar pengguna (*user*) untuk berhubungan atau berhubungan ke *website* yang sama dalam waktu yang bersamaan, suatu *web server* seringkali dikonfigurasi untuk melewatkan hubungan perangkat *client* hingga dalam jumlah yang maksimum, sehingga mengakibatkan perangkat *web server* sasaran menolak hubungan dari perangkat *client* berikutnya. Tentu saja, hal ini akan menyebabkan terjadinya suatu penolakan layanan *(Denial of Service*). Metode ini merupakan metode yang umum digunakan oleh para *hackers.*

Sejauh ini, penyerang tidak memiliki perangkat-perangkat yang digunakan untuk melakukan ancaman keamanan DDoS. Pengelola yang sebenarnya (*actual owners*) seringkali tidak perduli dengan sistem yang dimiliki, sehingga dapat digunakan sebagai perantara dalam melancarkan suatu ancaman keamanan DDoS. Penyerang umumnya menyebarkan *Trojan* yang terdiri dari beberapa kode berbahaya (*malicious*

code) yang dapat membuat seorang penyerang (*attacker*) dapat mengendalikan sistem yang dimiliki. *Malicious code* tersebut dapat juga dianggap sebagai suatu pintu belakang (*backdoor*). Sekali saja *Trojan* dijalankan, maka dapat menggunakan *email* untuk menginformasikan ke penyerang bahwa sistem dapat dikendalikan dari jarak jauh (*remotely controlled*). Penyerang akan meng-*install* perangkat-perangkat (*tools*) yang dibutuhkan untuk menampilkan suatu ancaman keamanan. Sekali seorang penyerang dapat mengendalikan sistem dengan baik, yang terkadang sistem yang sudah dikendalikan ini dinamakan dengan *zombie* atau *slaves*, maka penyerang akan dengan mudah melancarkan ancaman keamanan. Dalam beberapa kasus, sangat sulit atau tidak mungkin untuk menghindari ancaman keamanan DDoS secara keseluruhan. Beberapa pengarah rute (*routers*), *firewall*, dan sistem pendeteksi paket informasi yang tidak diinginkan atau *Intruder Detection Sistem (IDS)* dapat mendeteksi ancaman keamanan DDoS dan memblok (menghalangi) hubungan yang mencurigakan sehingga dapat mencegah layanan menjadi kelebihan beban (*overloaded*). Pada saat suatu perangkat atau perangkat jaringan telekomunikasi menjadi sasaran ancaman keamanan DDoS, penyedia layanan internet atau *Internet Service Provider (ISP)* dapat memblok alamat IP yang terlihat sebagai sumber dari ancaman keamanan. Sejauh ini, penyerang menyembunyikan alamat sumber yang sebenarnya, yang membuatnya menjadi sulit untuk melacak keberadaan si penyerang tersebut.

Jenis ancaman keamanan DDoS hampir sama dengan ancaman keamanan DoS, hanya saja, efek yang ditimbulkan lebih dahsyat, dibandingkan DoS, karena tidak hanya

melibatkan perangkat penyerang sebagai sumber ancaman keamanan, tetapi juga melibatkan banyak perangkat lain, yang tidak tahu menahu tentang ancaman keamanan DoS yang dilancarkan. Perangkat-perangkat lain yang berada dalam suatu sistem jaringan telekomunikasi dan digunakan sebagai perantara ancaman keamanan dapat dianggap juga sebagai sasaran, karena perangkat-perangkat tersebut telah disisipkan *Trojan*, yang tidak hanya sekedar dimanfaatkan untuk mengendalikan perangkat perangkat tersebut untuk melancarkan ancaman keamanan DoS secara bersama-sama, tetapi *Trojan* tersebut dapat juga dimanfaatkan untuk keperluan lain, seperti misalnya mengintip informasi perangkat sasaran, atau mengetahui apa saja yang dilakukan seorang sasaran terhadap perangkatnya. Selain itu, apabila sumber utama ancaman keamanan DDoS tidak dapat ditemukan secara pasti, pencarian atau investigasi sumber ancaman keamanan akan mengarah pada perangkat-perangkat perantara yang sering disebut dengan *zombie* ini. Akibatnya dapat ditebak, perangkat-perangkat perantara inilah yang kemudian dianggap sebagai pengirim ancaman keamanan DoS atau paling tidak, terlibat dalam ancaman keamanan DoS.

Ancaman keamanan DDoS mengandalkan kinerja *Trojan* yang disisipkan ke perangkat *zombie*. Oleh karena itu, perlu mewaspadai adanya atau masuknya *Trojan* ke perangkat atau perangkat komunikasi lainnya, pada saat perangkat atau perangkat komunikasi tersebut dihubungkan ke jaringan telekomunikasi, baik itu jaringan LAN, WLAN (WiFi), WAN, ataupun internet. Untuk itu, seorang pengguna perlu juga meng-install dan mengaktifkan program *Anti Trojan* dalam perangkat atau perangkat lain, seperti *handphone* atau PDA

yang digunakan untuk berkomunikasi.

Terdapat dua metode pendeteksian adanya ancaman keamanan DDoS yang menggunakan bentuk ancaman keamanan *TCP SYN Flood Attack*, yaitu:

1. *Sequential Methode Detection (SMD)*
   Merupakan sistem pendeteksian dalam suatu lokal jaringan telekomunikasi, dengan dua fase pendeteksian, yaitu fase pengetesan urutan atau S*equential Test Method (STM)* dan fase untuk mengawasi adanya alamat IP yang baru. Dua fase tersbut akan memberitahukan pengelola jaringan telekomunikasi apabila menemukan beberapa paket informasi yang melewati ambang batas kewajaran.
2. *Global Detector*
   Merupakan sistem pendeteksian pada jaringan telekomunikasi secara menyeluruh.

6.5. Serangan Penolakan Layanan Berlanjut

Ancaman serangan penolakan layanan berlanjut/*Pulse Denial of Service* (PdoS) pada aliran paket informasi menggunakan protokol transpor informasi TCP. Ancaman keamanan ini sangat efektif dibandingkan dengan DoS dan DDoS, karena meskipun memiliki kecepatan ancaman keamanan rata-rata lebih kecil, kerusakan sistem jaringan telekomunikasi yang ditimbulkan kurang lebih sama dengan DoS ataupun DDoS. Ancaman keamanan ini seringkali mengirimkan suatu urutan *(sequence)* ancaman keamanan yang meningkatkan kegiatan *router* sasaran (target), dan penyilangan aliran TCP akan membuat paket-paket informasi hilang secara periodik. Dengan demikian, secara signifikan, akan terjadi penurunan tingkat keluaran *(throughput*

degradation).

Ancaman keamanan PDoS ini membatasi suatu pengirim TCP untuk mengakses jaringan telekomunikasi sehingga berada dalam kondisi terlempar keluar *(timeout state)* dari jaringan telekomunikasi. Hal ini dapat dilakukan oleh penyerang dengan mengirimkan ancaman keamanan berdenyut *(pulse*) dengan pemilihan waktu secara instan. Selanjutnya, ancaman keamanan pengurangan kualitas atau R*eduction Of Quality (ROQ*) mengirimkan ancaman keamanan berdenyut secara periodik untuk mendorong *router* sasaran mengaktifkan mekanisme pengelolaan antrian (*queue management mechanism*) sehingga *router* sasaran memasuki kondisi sementara *(transient state)*. Pada akhirnya, ancaman keamanan PDoS menggunakan ancaman keamanan berdenyut (*pulses*) yang dapat menyebabkan pengirim TCP sasaran atau target ancaman keamanan mengalami kebuntuan (*congestion)* dalam mengakses jaringan telekomunikasi, sehingga akan memutuskan frekuensi sinyal yang dikirimkan.

Lebih lanjut, terdapat variasi ancaman keamanan PDoS, yaitu ancaman keamanan *Polymorphic* DoS *(PM*DoS*)*. Ancaman keamanan PMDoS adalah urutan denyut *(pulse)* ancaman keamanan. Setiap denyut *(pulse)* ancaman keamanan berakhir pada waktu periodik yang singkat ($T_{on} > 0$).dan intensitasnya dinyatakan oleh R_a dengan satuan *bit perseconds (bps)*. Dua pulsa yang berdekatan dipisahkan oleh suatu jarak (*interval*), $T_{off} \geq 0$. Secara umum, T_{on}, T_{off}, R_a dapat mengasumsikan nilai-nilai yang dapat diterima. Sejauh ini, untuk memfasilitasi pembahasan tentang PMDoS, konstanta R_a harus dapat ditentukan. Dengan catatan, PMDoS merupakan kasus khusus dalam ancaman keamanan PDoS. Ancaman

keamanan PMDoS dapat setara (*equivalent*) dengan ancaman keamanan PDoS atau ancaman keamanan ROQ pada saat semua nilai T_{on} dan T_{off} konstan. Jika T_{off} tertutup untuk satu detik dan T_{on} mendekati waktu lintasan atau *Round Trip Time (RTT)* pada aliran TCP sasaran, PMDoS dapat dianggap sebagai PDoS biasa. Meskipun demikian, apabila T_{off} mendekati nilai 0, efek ancaman keamanan PMDoS setara dengan ancaman keamanan DoS pada umumnya.

Dua tahapan sistem pendeteksian ancaman keamanan *Pulse Denial of Service (P*DoS*)* pada sisi penerima, dimana pendeteksian berdasarkan pada kehadiran dua anomali lalulintas paket informasi yang dibuat oleh suatu ancaman keamanan, dengan periode yang berubah-ubah (*fluctuation*) dalam lalulintas paket informasi TCP yang datang, dan kecenderungan penurunan lalulintas paket informasi TCP yang ditandai dengan ACK (*acknowledgement*). Pada tahapan yang pertama, sistem memantau atau mengawasi informasi yang datang dan lalulintas paket informasi ACK yang pergi menggunakan *discrete wavelet transform*. Pada tahapan yang kedua, terdapat algoritma non parameter CUSUM untuk mendeteksi ancaman keamanan PDoS dengan interval ancaman keamanan yang konstan. Sejauh ini, kedua tahapan tersebut dapat mendeteksi ancaman keamanan *Flood* DoS *(FD*DoS*)* atau PDoS dengan efektifitas yang sama karena ancaman keamanan akan menyebabkan periode yang berfluktuasi dalam lalulintas paket informasi TCP.

Seorang penyerang dapat mengirimkan pulsa-pulsa yang tidak digunakan (*useless pulses*) dalam paket informasi TCP pada suatu ancaman keamanan PMDoS. Penyerang tidak membutuhkan hubungan TCP yang tetap untuk melancarkan

suatu ancaman keamanan. Dengan perangkat *Vanguard*, ancaman keamanan PMDoS dapat dideteksi dari sisi penerima TCP dengan menganalisa lalulintas paket informasi TCP yang datang dan pergi menggunakan tanda ACK. *Vanguard* dirancang untuk mendeteksi ancaman keamanan yang datang ke beberapa *host* yang ditempatkan di belakangnya (jaringan lokal). *Host-host* tersebut menjalankan aplikasi TCP pada *client* untuk menerima informasi dari jaringan luar. *Vanguard* mendeteksi ancaman keamanan PMDoS berdasarkan pada jenis lalulintas paket informasi yang bersifat anomali, yaitu: anomali lalulintas paket informasi terlihat (*observable*), dan tidak dapat terlihat (*unobservable*).

6.6. Packet Sniffing

Packet sniffing adalah penyadapan terhadap lalu lintas data pada suatu jaringan telekomunikasi. *Packet sniffing* dikenal juga sebagai *Network Analyzers* atau *Ethernet Sniffer*, yaitu suatu aplikasi untuk memantau lalu lintas informasi dalam jaringan telekomunikasi. Dikarenakan data mengalir secara bolak-balik pada jaringan telekomunikasi, aplikasi ini menangkap setiap paket data yang lewat dan terkadang menguraikan isi *Request for Comments* (RFC) atau spesifikasi lainnya. Dengan menggunakan perangkat jaringan telekomunikasi, salah satu pihak dapat menyadap keseluruhan atau sebagian informasi dari perangkat yang terhubung dalam jaringan telekomunikasi. Perangkat pengendali jaringan telekomunikasi dapat pula diatur oleh aplikasi penyadap untuk bekerja dalam mode campur-aduk (*promiscuous mode*) untuk "mendengarkan semuanya".

Penyadapan atau *sniffing* ini dapat dibagi menjadi dua,

yaitu: *sniffing* pasif dan *sniffing* aktif. *Sniffing* pasif melakukan penyadapan tanpa mengubah data atau paket apapun di jaringan telekomunikasi, sedangkan *sniffing* aktif melakukan tindakan-tindakan atau perubahan paket data di jaringan telekomunikasi. s*niffing* pasif dapat ditanggulangi dengan cara menggunakan *switch*. Sedangkan pada *sniffing* aktif, *Address Resolution Protocol* (ARP) cache dimodifikasi sehingga dapat membelokkan data dari perangkat sasaran ke perangkat *hacker*. ARP adalah suatu protokol dalam TCP/IP Protocol Suite yang bertanggungjawab dalam melakukan resolusi alamat IP ke dalam *alamat Media Access Control* (*MAC Address*). ARP didefinisikan dalam RFC 826.

Pada kenyataannya, masih sedikit solusi yang tepat untuk mendeteksi maupun untuk mencegah *snffing* ini. Sistem deteksi penyusup jaringan telekomunikasi yang ada saat ini umumnya mampu mendeteksi berbagai jenis serangan tetapi tidak mampu mengambil tindakan lebih lanjut.

Packet Sniffing dapat dilakukan dengan memantau atau menganalisa paket data yang ditransmisikan dari perangkat *client* ke perangkat web *server*. *Tool* yang biasa digunakan untuk melakukan teknik *packet sniffing* ini biasanya adalah *Wireshark* dan *Netcut*. Packet sniffing ini biasanya dilakukan oleh para hacker atau penyusup yang berbahaya untuk melakukan tindakan yang dilarang seperti mencuri *password*, dan pengambilan data-data penting lainnya.

Cara kerja *packet sniffing* dibagi menjadi tiga, yaitu:

1. Pengumpulan (*Collecting*)

   Cara kerja yang pertama dari *packet sniffing* adalah mengubah interface yang digunakan menjadi "*promicius mode*", dan mulai mengumpulkan atau mengelompokkan

semua paket data yang lewat melalui jaringan lokal bentuk *raw binary*.

2. Pengubahan (*Conversion*)
   Cara kedua adalah mengkonversi atau mengubah data yang berbentuk *binary* kedalam data yang mudah dibaca atau mudah dipahami.
3. Analisa (*Analysis*)
   Cara kerja ketiga adalah dimana betuk data tersebut diklasifikasikan kedalam blok-blok protocol berdasarkan sumber dari transmisi data tersebut baik berupa tcp, udp dan lain-lain.
4. Pengambilan atau Pencurian Data
   Cara kerja yang terakhir adalah setelah melakukan klasifikasi terhadap data-data yang telah dikirim, maka hacker atau penyusup melakukan pencurian data.

Untuk menangani atau mengatasi *packet sniffing* ini adalah dengan melakukan enkripsi data yang ingin dikirim melalui jaringan telekomunikasi. Untuk meminimalkan atau mengurangi *packet sniffing* dengan mengganti semua hub dengan switch. Selain itu, dapat juga menggunakan protokol-protokol yang memiliki standar yang aman, misalnya penggunaan protokol-protokol yang sudah dilengkapi dengan enkripsi data untuk mengamankan data atau informasi, seperti: IPSec, SMB Signing, HTTPS, dan lain sebagainya.

6.7. IP Spoofing

IP Spoofing adalah pengubahan alamat IP pada perangkat komunikasi atau perangkat penyerang dengan alamat IP yang dimiliki oleh perangkat komunikasi atau perangkat target ancaman keamanan (sasaran), sehingga seolah-olah perangkat

komunikasi atau perangkat target ancaman keamanan (sasaran) merupakan asal (sumber) permintaan PING ke sistem jaringan telekomunikasi. Dengan menggunakan IP *spoofing*, tanggapan dari PING tersebut kemudian dialamatkan ke perangkat yang IPnya sudah di-*spoof* tersebut. Akibatnya, perangkat komunikasi atau perangkat target ancaman keamanan (sasaran) akan menerima banyak paket informasi tanggapan permintaan IP. Dapat dibayangkan apabila perangkat yang di-*spoof* tersebut memiliki hubungan berkecepatan rendah dan PING diarahkan ke sistem jaringan telekomunikasi yang memiliki banyak *host*. Hal ini dapat menyebabkan terjadinya *hang* pada perangkat komunikasi atau perangkat target ancaman keamanan (sasaran) serta sistem jaringan telekomunikasi yang menghubungkan perangkat komunikasi atau perangkat target ancaman keamanan (sasaran) dengan perangkat-perangkat lainnya yang menerima permintaan IP (*IP Request*).

BAB VII
KEAMANAN JARINGAN BERBASIS PROTOKOL KOMUNIKASI

7.1. Pengamanan Berbasis OSI Layer

Untuk dapat memahami keamanan jaringan telekomunikasi, terlebih dahulu memahami aturan/tata cara berkomunikasi antara suatu perangkat dengan perangkat lainnya yang dihubungkan oleh suatu media transmisi. Aturan/tata cara berkomunikasi dapat disebut sebagai protokol komunikasi. *Open System Interconnection* (OSI) adalah protokol komunikasi yang distandarisasi oleh Badan Standar Internasional (*International Standard Organization*) yang dibagi menjadi tujuh tahapan atau tujuh lapisan (*seven layers*), yaitu:

1. Lapisan 1: Lapisan Fisik (*Physical Layer*)
   Lapisan atau tahapan komunikasi ini berkaitan dengan perangkat komunikasi dan media transmisi. Perangkat komunikasi dibedakan atas: terminal atau *Data Terminal Equipment (DTE)* dan penghubung atau *Data Communication Equipment (DCE).* Wujud dari terminal dapat berupa perangkat, tablet, smartphone dan lain sebagainya, baik yang berfungsi sebagai *client* (penerima layanan) maupun *server* (pengatur layanan). Wujud dari penghubung di antaranya adalah konektor-konektor, *hub, switch, bridge, router, modem* dan sebagainya. Untuk media transmisi dapat dibedakan atas: media transmisi dengan kabel (*on wire*) dan tanpa kabel (*wireless*). Untuk

saat ini, media transmisi dengan kabel dapat berupa kabel terbuka (open wire), kabel pasangan terpilin (twisted pair wire), kabel koaksial (coaxial wire) dan kabel serat optik (fiber optic cable). Sedangkan media transmisi tanpa kabel dapat berupa: sarana yang menggunakan infra merah, ultra violet, gelombang radio, gelombang cahaya.

2. Lapisan 2: Lapisan Hubungan Data (*Data Link Layer*)
   Lapisan ini mengatur bagaimana data yang akan dikirimkan diubah menjadi deretan angka 1 dan 0 dan mengirimkannya ke media fisik. Sedangkan pada sisi penerima, lapisan ini mengubah angka 1 dan 0 yang diterima dari media fisik menjadi data yang lebih berarti. Dalam lapisan ini juga mengatur dan mengatasi kesalahan-kesalahan yang mungkin terjadi ketika transmisi data dilakukan.
3. Lapisan 3: Lapisan Jaringan telekomunikasi (*Network Layer*)
   Lapisan ini bertanggungjawab terhadap hubungan antara pengirim dan penerima, dan menentukan *routing* (pengarahan sinyal) berdasarkan alamat pengirim (*source address*) dan alamat penerima (*destination address*). Suatu *router* akan menentukan rute terpendek yang efisien. Penentuan rute ini dapat dilakukan secara statis maupun dinamis.
4. Lapisan 4: Lapisan Pengangkutan (*Tranport Layer*)
   Lapisan ini bertanggung jawab untuk mengatur pengangkutan data yang bebas dari gangguan. Terdapat dua macam metode pengangkutan, yaitu: yang berorientasi pada hubungan (*connection oriented*), misalnya: *Transmission Control Protocol (TCP)* dimana dipastikan tidak ada gangguan dalam pengangkutan data, dan jika

terjadi gangguan, terdapat mekanisme untuk pengiriman ulang data yang rusak karena gangguan, dan yang tidak berorientasi pada hubungan (*connectionless*), misalnya: *User Datagram Protocol (UDP)* yang tidak memiliki mekanisme untuk memastikan apakah terdapat gangguan terhadap data yang dikirimkan, dan tidak dapat memastikan data yang sampai di penerima masih dalam kondisi baik

5. Lapisan 5: Lapisan Sesi (*Session Layer*)
   Lapisan ini bertanggungjawab untuk membangun, memlihara dan menutuskan hubungan antar aplikasi, dan menentukan berapa lama waktu yang disediakan untuk melaksanakan hubungan.
6. Lapisan 6: Lapisan Presentasi (*Presentation Layer*)
   Lapisan ini bertanggungjawab mengatur format data yang digunakan untuk menampilkan aplikasi di lapisan atasnya dan dapat diterima di lapisan bawahnya. Dengan kata lain, lapisan ini bertanggungjawab terhadap pengkodean data, enkripsi dan dekripsi data.
7. Lapisan 7: Lapisan Aplikasi (*Application Layer*)
   Lapisan ini adalah lapisan yang terhubung langsung ke pengguna dan menampilkan layanan yang diperlukan/digunakan oleh pengguna. Lapisan ini dapat berupa perambah situs (*browser*), pembuka *email* dan sebagainya.

Ditinjau dari jenis perangkat, umumnya lapisan OSI 7 *layer* ini diklasifikasikan menjadi 3 jenis, yaitu:

1. Lapisan yang dikelola dengan perangkat keras
   Lapisan ini berada di lapisan bawah, yaitu lapisan fisik, lapisan hubungan data, dan lapisan jaringan telekomunikasi.

2. Lapisan perantara (dapat dikelola perangkat keras dan perangkat lunak)
   Lapisan ini adalah lapisan tengah, yaitu lapisan pengangkutan.
3. Lapisan yang dikelola dengan perangkat lunak
   Lapisan ini berada di lapisan atas, yaitu lapisan sesi, lapisan presentasi, dan lapisan aplikasi.

Dari protokol standar tersebut, dalam kenyataannya, tidak semuanya dilakukan, misalnya pada protokol X-25, hanya menggunakan lapisan yang dikelola perangkat keras. Sedangkan pada protokol yang digunakan dalam internet, yaitu protokol Transmission Control Protocol/Internet Protocol (TCP/IP), lapisan yang dikelola dengan perangkat lunak dijadikan satu menjadi lapisan aplikasi (*Application Layer*), lapisan berikutnya, yaitu lapisan pengangkutan menggunakan protokol TCP (*TCP layer*), lapisan jaringan telekomunikasi (*IP layer*), lapisan hubungan data dan lapisan fisik dijadikan satu menjadi lapisan antarmuka jaringan (*network interface layer*).

7.2. Pengamanan Berbasis Protokol 802.x

Dikarenakan perbedaan fungsi dalam setiap lapisan jaringan, maka perlindungan yang dilakukan juga berbeda-beda. Terdapat dua mekanisme yang digunakan dalam mengamankan titik akses ke jaringan telekomunikasi, yaitu:

1. Pengesahan berbasis protokol 802.1x
   Merupakan suatu protokol yang dapat melakukan otentifikasi pengguna dari peralatan yang melakukan hubungan ke suatu titik akses. Dengan protokol ini, ketika suatu terminal melakukan hubungan ke perangkat penghubung, pengguna perlu melakukan pengesahan

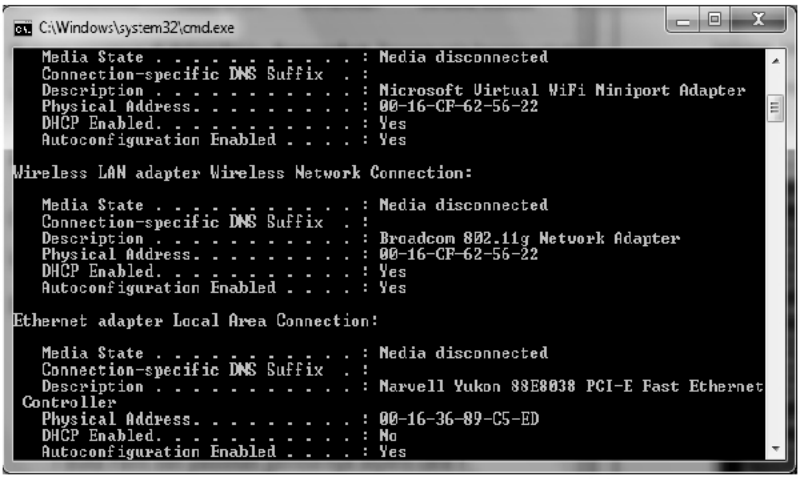
(*autentification*) sebelum terhubung ke jaringan telekomunikasi. Protokol ini melibatkan pengakses (terminal), *server*, dan perangkat penghubung, dengan tahapan-tahapan sebagai berikut:

a. Secara *default*, seorang pengguna melakukan akses ke jaringan telekomunikasi dengan meminta ijin akses ke *server.*
b. *Server* memberi pertanyaan-pertanyaan yang harus dijawab dengan benar oleh pengguna.
c. Jika jawaban benar, maka pengguna dapat mengakses jaringan telekomunikasi, dan jika salah,maka pengguna mengulangi permintaan akses ke *server*.

2. Pengesahan berbasis *Medium Access Control Address (MAC Address*)

   Pengesahan berbasis *MAC Address* adalah suatu mekanisme dimana suatu peralatan yang akan melakukan akses pada suatu titik akses, terdaftar terlebih dahulu. Berbeda dengan mekanisme pengesahan berbasis protokol 802.1x yang memastikan bahwa terminal yang melakukan hubungan digunakan oleh pengguna yang sudah disahkan *server* (terdaftar berbasis pengguna), maka pada pengesahan berbasis *MAC Address* memastikan bahwa apakah terminal yang akan melakukan akses sudah terdaftar alamat fisiknya (alamat MAC) di *server* (terdaftar berbasis terminal yang digunakan). Alamat fisik selalu menyertai perangkat yang menjadi terminal, umumnya sudah dicantumkan dalam kartu antarmuka jaringan (*network interface card*) pada saat pembuatan kartu antarmuka jaringan tersebut. Untuk mengetahui *MAC Address* atau *Physical Address*, dapat mengetikkan perintah

ipconfig/all di c-prompt, seperti dalam Gambar 7.1. berikut.



Gambar 7.1. Tampilan MAC Address

Dalam contoh di atas, *physical address* WLAN Card adalah: 00-16-cf-62-56-22, sedangkan *physical address* LAN Card adalah: 00-16-36-89-C5-ED

Terdapat kelebihan dan kekurangan pada kedua metode tersebut di atas, dimana kelebihan pengesahan berbasis *MAC Address* dibandingkan pengesahan berbasis protokol 802.1x adalah lebih banyak diterapkan pada *switch/hub* yang sering digunakan sebagai titik akses dimana tidak perlu semua *switch/hub* melakukan *filtering*, sedangkan kelemahan pengesahan berbasis *MAC Address* dibandingkan pengesahan berbasis protokol 802.1x adalah seseorang dengan mudah memanipulasi identitas unik terminal sehingga dapat mengakses jaringan telekomunikasi secara ilegal.

BAB VIII
FIREWALL

8.1. Pengertian dan Perbandingan Firewall

Firewall merupakan suatu perangkat yang diletakkan antara internet dengan jaringan lokal. Informasi yang keluar atau masuk harus melalui *firewall* ini. Tujuan adanya *firewall* adalah untuk mencegah (*prevent*) agar akses (ke dalam maupun ke luar) dari orang yang tidak berwenang (*unauthorized access*) tidak dapat dilakukan.

Firewall bekerja dengan mengamati paket IP (*Internet Protocol*) yang melewatinya. Berdasarkan pada konfigurasi *firewall* maka akses dapat diatur berdasarkan *IP Address*, *port*, dan arah informasi. Detail konfigurasi bergantung pada masing-masing *firewall* dan kebijaksanaan (*policy*) organisasi yang bersangkutan, yang dapat dibagi menjadi dua jenis:

1. *Prohibitted*, yaitu kebijakan melarang akses tertentu dari jaringan telekomunikasi ke internet.
2. *Permitted*, yaitu kebijakan mengijinkan jaringan telekomunikasi untuk mengakses internet.

Secara konseptual terdapat 2 macam *firewall*:

1. *Network Firewall*
   Keputusan mengijinkan atau melarang akses ke internet berdasarkan pada alamat sumber, alamat tujuan dan *port* yang terdapat dalam setiap paket IP.
2. *Application Firewall*
   Keputusan mengijinkan atau melarang akses ke internet berdasarkan pada *host* yang berjalan sebagai *proxy server*,

yang tidak mengijinkan lalulintas antar jaringan telekomunikasi dan melakukan *logging* dan *auditing* lalulintas yang melaluinya.

Firewall dapat berupa suatu perangkat keras yang sudah dilengkapi dengan perangkat lunak tertentu, sehingga pengguna (administrator) selanjutnya melakukan konfigurasi dari *firewall* tersebut. *Firewall* juga dapat berupa perangkat lunak yang ditambahkan ke suatu *server*, yang dikonfigurasi menjadi *firewall*.

Untuk menjaga fungsi komunikasi jaringan lokal lingkungan yang dipasang *firewall*, dilakukan dua cara:

1. *Packet filtering*, yaitu: mekanisme pengontrolan informasi yang diperbolehkan mengalir dari dan atau ke jaringan lokal dengan menggunakan beberapa paremeter yang tercantum dalam header paket informasi: arah (*inbound* atau *inbound*), alamat asal dan tujuan, *port* asal dan tujuan serta jenis protokol transport. seperti *telnet* dan SMTP (*Single Mail Transport Protocol*).
2. *Proxy system*, dimana setiap komunikasi yang terjadi antara dua jaringan telekomunikasi harus dilakukan melalui suatu operator, dalam hal ini *proxy server*.

Protokol *File Transport Protocol* (FTP) lebih efektif ditangani dengan sistem *proxy*. Kebanyakan *firewall* menggunakan kombinasi kedua teknik ini (*packet filtering* dan *proxy*). Beberapa perangkat lunak berbasis UNIX yang dapat digunakan untuk melakukan IP *filtering* antara lain:

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

8.2. Network Firewall

Network firewall yang pertamakali muncul pada akhir era 1980-an yaitu berupa perangkat *router* yang dipakai untuk memisahkan suatu jaringan telekomunikasi menjadi jaringan lokal (LAN) yang lebih kecil, dimana kondisi ini penggunaan *firewall* hanya dimaksudkan untuk mengurangi masalah peluberan (*spillover*) data dari LAN ke seluruh jaringan telekomunikasi untuk mencegah masalah masalah semacam error pada pengelolaan jaringan telekomunikasi, atau aplikasi yang terlalu banyak menggunakan sumber daya meluber ke seluruh jaringan telekomunikasi. Penggunaan *firewall* untuk keperluan keamanan (*security firewall*) pertamakali digunakan pada awal dekade 1990-an, berupa *router* IP dengan aturan *filter* tertentu. Aturan keamanan saat itu berupa sesuatu seperti: ijinkan setiap orang "disini" untuk mengakses "keluar sana", juga cegahlah setiap orang (atau apa saja yang tidak disukai) "di luar sana" untuk masuk "kesini". *Firewall* semacam ini cukup efektif, tetapi memiliki kemampuan yang terbatas. Seringkali sangat sulit untuk menggunakan aturan *filter* secara benar. Sebagai contoh, dalam beberapa kasus terjadi kesulitan dalam mengenali seluruh bagian dari suatu aplikasi yang dikenakan restriksi. Dalam kasus lainnya, aturan *filter* harus diubah apabila ada perubahan "diluar sana".

Firewall generasi selanjutnya yang lebih fleksibel, yaitu berupa *firewall* yang dibangun pada "*Bastion Host*". *Firewall* komersial yang pertama dari jenis ini, yang menggunakan *filter* dan *gateway* aplikasi (proxies), kemungkinan adalah produk dari Digital Equipment Corp (DEC). DEC yang dibangun berdasarkan *firewall* korporat DEC. Brian Reidd di laboratorium sistem jaringan telekomunikasi DEC di Pallo Alto

adalah pencipta *firewall* DEC.

Firewall komersial pertama dikirimkan ke pelanggan pertamanya, suatu perusahaan kimia besar yang berbasis di pantai timur AS pada 13 Juni 1991. Dalam beberapa bulan kemudian, Marcus Ranum dari Digital Corp menciptakan security proxies dan menulis ulang sebagian besar kode program *firewall*. Produk *firewall* tersebut kemudian diproduksi missal dengan nama dagang DECSEAL (singkatan dari Security External Access *Link*). DECSEAL tersusun atas suatu sistem luar yang disebut gatekeeper sebagai satu satunya sistem yang dapat berhubungan dengan internet, suatu *filter*ing *gateway* yang disebut gate, dan suatu mailhub.

8.3. Application Firewall

Firewall ini disebut juga sebagai *firewall* berbasis *proxy* yang beroperasi di level aplikasi dan dapat mempelajari informasi pada level data aplikasi (yang dimaksudkan disini adalah isi (*content*) dari paket data karena *proxy* pada dasarnya tidak beroperasi pada paket data). Filterisasi dilakukan berdasarkan data aplikasi, seperti perintah-perintah FTP atau URL yang diakses lewat HTTP. Dapat dikatakan bahwa *firewall* jenis ini memecah model *client server*. Cara kerjanya adalah apabila ada pengguna yang menggunakan salah satu aplikasi semisal FTP untuk mengakses secara *remote*, maka *gateway* akan meminta pengguna memasukkan alamat *remote host* yang akan diakses. Saat pengguna mengirimkan *User ID* serta informasi lainnya yang sesuai maka *gateway* akan melakukan hubungan terhadap aplikasi tersebut yang terdapat pada *remote host*, dan menyalurkan data diantara kedua titik. Apabila data tersebut tidak sesuai maka *firewall* tidak akan

meneruskan data tersebut atau menolaknya. Lebih lanjut, pada *firewall* jenis ini dapat dikonfigurasikan untuk hanya mendukung beberapa aplikasi saja dan menolak aplikasi lainnya untuk melewati *firewall*. Kelebihannya adalah relatif lebih aman daripada jenis *packet filtering*. *Router* lebih mudah untuk memeriksa (*audit*) dan mendata (*log*) semua aliran data yang masuk pada level aplikasi. Kekurangannya adalah pada pemrosesan tambahan yang berlebih pada setiap hubungan. Hal ini akan mengakibatkan adanya dua sambungan antara pengguna dan *gateway*, dimana *gateway* akan memeriksa dan meneruskan semua arus dari dua arah.

8.4. Arsitektur Firewall

Arsitektur *Firewall* dapat dibedakan atas:

1. *Dual-Homed Host* (*Dual Homed Gateway/DHG*)
   Menggunakan suatu perangkat dengan (minimal) dua NIC. Interface pertama dihubungkan ke jaringan lokal dan yang lainnya dengan internet. *Dual Homed host*-nya sendiri berfungsi sebagai *bastion host* (Suatu sistem perangkat yang harus memiliki keamanan jaringan telekomunikasi yang tinggi, karena biasanya peka terhadap ancaman keamanan jaringan telekomunikasi.
2. *Screened-Host (Screened Host Gateway/SHG)*
   Fungsi *firewall* dilakukan oleh suatu *screening-router* dan *bastion host*. *Router* ini akan menolak semua trafik kecuali yang bertujuan ke *bastion host*, sedangkan pada trafik dalam tidak dilakukan pembatasan.
3. *Screened Subnet* (*Screened Subnet Gateway*/SSG)
   Firewall dengan arsitektur ini menggunakan dua *Screened-router* dan jaringan telekomunikasi tengah (*perimeter*

network) antara kedua *router* tersebut, dimana ditempatkan *bastion host*. Beberapa Perangkat lunak *Firewall*, Zone Alarm Pro *Firewall,* PC Tools *Firewall* Plus, *Norton Internet Security*.

BAB IX
SISTEM PENGAMANAN JARINGAN

9.1. Intrusion Detection System (IDS)

Intrusion Detection System (IDS) adalah suatu perangkat lunak berbasis *host* pada jaringan telekomunikasi yang mendeteksi adanya percobaan penyusupan data oleh pihak asing (yang tidak disahkan oleh sistem *authentification* jaringan telekomunikasi). IDS menerima salinan paket data yang bertujuan pada suatu *server* untuk kemudian memeriksa paket-paket data tersebut. Apabila ditemukan adanya paket data yang dideteksi sebagai suatu ancaman terhadap keamanan jaringan telekomunikasi maka IDS memberi peringatan pada administrator jaringan telekomunikasi. Karena IDS hanya memeriksa dan mendeteksi salinan paket data, sekalipun ditemukan adanya paket data yang terdeteksi sebagai ancaman terhadap keamanan jaringan telekomunikasi, paket data tersebut tetap sampai ke *server* tujuan. Dalam hal ini, IDS bersifat pasif.

Metode yang digunakan dalam IDS adalah:

1. *Signature based Intrusion Detection System*
   Dalam metode ini, terdapat daftar tandatangan (*signature*) yang dapat digunakan untuk menilai apakah paket data dapat menjadi ancaman terhadap keamanan jaringan telekomunikasi, dengan mencocokkan tandatangan pada paket data tersebut dengan daftar paket data yang sudah ada, yang diperbarui secara periodik.Metode ini melindungi sistem dari jenis-jenis seragangan yang sudah

diketahui sebelumnya.

2. *Anomaly based Intrusion Detection System*

   Dalam metode ini, pengelola jaringan telekomunikasi melakukan konfigurasi sehingga mengetahui paket data apa saja yang dpat memasuki suatu jaringan telekomunikasi. Suatu paket anomali adalah paket yang tidak sesuai dengan kebiasaan jaringan telekomunikasi. Apabila IDS menermukan adanya anomaly pada paket data yang diterima atau dikirimkan, maka IDS memberikan pemberitahuan pada administrator jaringan telekomunikasi. Untuk itu, administrator jaringan telekomunikasi harus memberitahu IDS secara terus menerus bagaimana lalu lintas data yang normal pada suatu sistem jaringan telekomunikasi untuk menghindari adanya salah pendeteksian oleh IDS.

Dalam penerapannya, IDS adalah suatu unit *host* yang terhubung pada suatu hub/switch dan akan menerima salinan dari paket-paket data yang diproses oleh hub/switch. IDS yang berbasis *host* akan memeriksa sistem panggilan, catatan dan perubahan sistem *file* pada *host* untuk mencari anomaly atau keanehan yang menandakan adanya usaha dari pihak luar untuk menyusup ke dalam sistem. IDS berbasis *host* akan membantu pengelola sistem melakukan *audit trail* terhadap sistem apabila terjadi penyusupan dalam sistem.

Pemantauan jaringan dapat dilakukan dengan menggunakan *Network Monitoring*, biasanya dilakukan dengan menggunakan protokol SNMP *(Simple Network Management Protoco*l). Contoh-contoh program *network monitoring/management* antara lain: *Etherboy*, *Etherman*, HP *Openview*, *Packetboy*, *Packetman*, SNMP Collector, *Webboy*.

Contoh program pemantau jaringan telekomunikasi yang tidak menggunakan SNMP antara lain: *iplog, icmplog, updlog,* yang merupakan bagian dari paket *iplog* untuk memantau paket IP, ICMP, UDP.

Sistem pemantauan jaringan menggunakan IDS dapat mengetahui adanya penyusup (*intruder*) atau adanya ancaman keamanan (*attack*). Sistem ini dapat memberitahu administrator melalui *email* maupun melalui mekanisme lain seperti melalui sms. Contoh perangkat lunak IDS antara lain:

1. *Autobuse*, mendeteksi *probing* dengan memantau log*file*.
2. *Portsentry,* mendeteksi *probing* (*port scanning*) dengan memantau paket data yang melewati *jaringan telekomunikasi* dan memasukkan IP penyerang dalam penyaringan.
3. *Snort*, mendeteksi pola (*pattern*) pada paket yang lewat dan mengirimkan pemberitahuan jika pola tersebut terdeteksi.
4. *Honeypot*, merupakan suatu sistem yang digunakan untuk memancing dan memantau *hacker*, berupa kumpulan perangkat lunak (*server*) yang seolah-olah merupakan *server* yang hidup dan memberi layanan SMTP yang memantau asal hubungan dan kegiatan penyerang.

9.2. Intrusion Prevention System (IPS)

Intrusion Prevention System (IPS) adalah suatu perangkat lunak berbasis *server* pada jaringan telekomunikasi yang dapat mendeteksi dan memberikan suatu keputusan apakah suatu paket data dapat diterima atau tidak oleh suatu *server* jaringan telekomunikasi. Dengan terhubung ke *firewall*, IPS dapat memberitahu *firewall* untuk menolak paket data yang

terdeteksi sebagai ancaman terhadap keamanan jaringan telekomunikasi

Dengan adanya IPS, maka waktu yang dibutuhkan suatu paket untuk mencapai *host* tujuan menjadi lama, dan tidak cocok untuk aplikasi-aplikasi yang membutuhkan pengiriman data dalam waktu nyata (*real time*). Selain itu, IPS dan juga IDS juga masih membuka kesempatan untuk terjadinya *false positive*, yaitu suatu paket data yang aman dinyatakan berbahaya dan *false negative*, yaitu paket data yang berbahaya dinyatkan aman. Untuk mengurangi terjadinya *false positive* dan *false negative* maka perlu dilakukan pembaruan secara rutin terhadap IPS dan juga IDS.

Dalam penerapannya, IPS ditempatkan pada unit yang sama dengan *firewall* dan memproses paket-paket data yang lewat melalui *firewall*.

9.3. Pengamanan Pada Jaringan IPv6

Sebagaimana halnya dengan jaringan internet yang menggunakan pengalamatan IPv4 (32 bit), IPv6 (128 bit) dapat juga mengalami ancaman keamanan dengan memanfaatkan ketidakamanan IPSecv6. Informasi-informasi yang dikirimkan melalui *Internet Control Message Protokol (ICMP)* dalam jumlah yang banyak, dapat membuat *server* sasaran menjadi rusak. Selain itu, dengan ICMP, seorang penyerang dapat mengendalikan suatu jaringan telekomunikasi untuk melancarkan ancaman keamanan DDoS.

Ancaman keamanan pada jaringan internet IPv6 dapat memanfaatkan prosedur DAD yang dapat mendeteksi kemungkinan adanya alamat IP yang sama dalam suatu jaringan telekomunikasi. Penyerang akan mengirimkan alamat

IP yang telah dimiliki oleh *node* lain yang berdekatan ke *node* yang menjadi target ancaman keamanan secara terus menerus, sehingga alamat *node* target, selalu sama dengan *node* lain yang berdekatan tersebut. Hal ini akan membuat prosedur DAD terus dijalankan untuk menghasilkan alamat IP yang baru, agar tidak ada alamat IP yang sama dalam suatu jaringan telekomunikasi. Jika prosedur DAD terus berjalan, maka jaringan telekomunikasi akan sibuk sehingga dapat menolak permintaan layanan yang datang.

Dalam menghadapi ancaman keamanan, dirancang suatu bentuk jaringan telekomunikasi yang dapat diprogram (*programmable networks*). Pada dasarnya, jaringan telekomunikasi yang dapat diprogram tersusun dari sejumlah *router* yang dapat diprogram, yang disebut dengan pemrograman titik simpul (*programmable node)* atau *active node* dalam suatu jaringan internet *Protocol (IP). Programmable node* mengidentifikasi paket khusus yang disebut dengan *active packet* dan memberikan suatu kode khusus untuk memprosesnya. *Active packet* berasal dari suatu sistem sumber akhir ke suatu sistem tujuan akhir, dimana *programmable node* berada dalam suatu bagian antara sumber dan tujuan pemrosesan *active packet.*

Rancangan jaringan telekomunikasi yang dapat diprogram meliputi jaringan telekomunikasi dengan banyak layanan (*multiservice network*) dan jaringan telekomunikasi dengan banyak penyedia layanan (*multidomain network*). Jaringan telekomunikasi dengan banyak layanan membuat pengguna jaringan telekomunikasi IP dapat menjalankan layanan pemrograman melalui *programmable node* yang berada dalam arsitektur jaringan telekomunikasi yang

dirancang tersebut. Pemrograman titik simpul (*programmable node*) menjalankan kode untuk memproses paket informasi yang aktif (*active packet*), yang dapat membawa informasi pengguna dan mengendalikan informasi.

Pada jaringan telekomunikasi yang dapat diprogram, bentuk jaringan telekomunikasi atau topologinya dapat berubah, sehingga dapat menghasilkan perubahan rute jaringan telekomunikasi dengan memanfaatkan titik simpul terprogram baru yang terlihat dalam suatu jaringan telekomunikasi atau pada saat *programmable node* tidak dapat berfungsi (*down*). Perubahaan topologi dapat menyebabkan perubahan pengambilan tempat secara tiba-tiba, sebagai permulaan *programmable node* yang baru dalam memproses *active packets* dari layanan terprogram atau pada saat *programmable node* lain menahan pemrosesan *active packet*. Arsitektur keamanan harus tahan *(immune)* terhadap perubahan topologi ini.

Dalam jaringan telekomunikasi *multiservice,* proses *authentication* perlu diverifikasi pada saat seorang pengguna meminta suatu layanan ke *multiservice network*, memproses suatu kode *server* ke *programmable node,* dan pada saat suatu *programmable node* menerima suatu *active packet*. *Programmable node* dapat memodifikasi sebagian *active packet*, yang disebut dengan *dynamic packet*. Hal ini membutuhkan adanya mekanisme keamanan baru, dimana hanya modifikasi yang telah dibuat oleh *programmable node* yang sah (*authorized)* saja yang dapat dilewatkan. Mekanisme ini merupakan layanan terpadu yang dapat meyakinkan bahwa *dynamic part* dari *active packet* tidak dapat dimodifikasi oleh *programmable node* yang tidak sah. Hal ini tentu saja

dilakukan untuk menghindari kemungkinan adanya ancaman keamanan DoS, yang dapat meracuni proses *authentication*, dengan terlebih dahulu mencuri *active packet*.

Pada pemrograman jaringan telekomunikasi dengan banyak penyedia layanan (*multidomain network)*, titik simpul terprogram terletak pada jaringan telekomunikasi dimana penyedia layanan atau *Internet Service Provider (ISP)* berada, yang biasanya terletak di luaran jaringan internet. ISP ini memberikan layanan secara langsung ke pengguna.

Proses klarifikasi sistem keamanan pada rancangan jaringan telekomunikasi dapat diuraikan sebagai berikut:

1. Pengguna meminta pengesahan (*authorization*) dari *Authorization Server*, dengan mengirimkan parameter C, SST, SET, D, S, U dan SP. Pengertian dari parameter-parameter tersebut adalah sebagai berikut:
   a. Parameter C atau *Code* adalah parameter yang mengidentifikasi kode yang dapat dijalankan (*executeable code*) yang harus memproses *active packet* untuk menawarkan layanan dalam *programmable node*. Karena setiap layanan yang dapat diprogram tergabung dengan *executeable code* yang berbeda, C_i merupakan nilai yang mengidentifikasi layanan yang dapat diprogram, yang dibutuhkan oleh seorang pengguna.
   b. Parameter SST atau *Service Start Time* adalah parameter waktu dimana layanan terprogram dimulai. *Programmable node* tidak harus memproses *active packet* yang datang sebelum waktu diindikasi atau ditentukan oleh parameter SST ini.
   c. Parameter SET atau *Service End Time* adalah parameter waktu dimana layanan terprogram berakhir.

Programmable node tidak harus memproses *active packet* yang datang setelah waktu diindikasi atau ditentukan oleh parameter SET ini.

d. Parameter D atau *Destination* adalah parameter alamat IP tujuan dari *active packet*, dan merupakan akhir dari suatu sistem komunikasi yang dilayani oleh jaringan telekomunikasi. *Active packet* harus diverifikasi di antara parameter S dan D atau di antara awal dan akhir sistem komunikasi.
e. Parameter S atau *Source* adalah parameter yang menyatakan alamat IP sumber dari *active packet* yang dikirimkan.
f. Parameter U atau Pengguna adalah parameter yang mengidentifikasi pengguna (pengguna) yang meminta layanan terprogram, yang ditanggapi untuk kebutuhan penggunaan layanan yang terprogram.
g. Parameter SP atau *Spesific Parameter* adalah parameter khusus yang tergantung pada tanggapan dalam layanan terprogram.*Programmable node* harus mengaplikasikan aturan pengesahan untuk *active packet* yang datang. Untuk membuat mungkin, parameter pengesahan harus hadir pada setiap *programmable node.*

2. *Authorization Server* sebagai pemberi pengesahan, menghasilkan *session key* dan mengirimkannya ke pengguna.
3. Pengguna menghasilkan suatu *active packet* dengan mengenalkan parameter pengesahan yang telah dikirimkan dan melindunginya dengan menggunakan *session key*. Akhirnya, pengguna akan mengirimkan *active packet* ke arah tujuan (D).

4. Pada saat *programmable node* menerima suatu *active packet*, yang mana tidak memiliki *execution code* yang diidentifikasi oleh paramter C, dan tergabung ke dalam kunci Kc_i, *programmable node* akan mengambilnya *(download)* dari *Code Server*. Lalu, *programmable node* menghasilkan *session key* dengan menggunakan kunci Kc_i dan mengesahkan parameter yang membawa *active packet*, serta memverifikasi *integrity* dan *authentication* dari *active packet* tersebut. *Programmable node* juga memverifikasi pengesahan untuk pemrosesan paket dengan menggunakan parameter-paramter pengesahan.
5. Sekali *active packet* diproses, jika kemudian dimodifikasi, *programmable node* akan melindungi *active packet* tersebut dengan menggunakan *session key*. Akhirnya, *programmable node* mengirimkan *active packet* ke arah tujuan.

Dalam skenario *multidomain network* terdapat sejumlah *domain*, dimana sejumlah *active packet* yang memasuki beberapa sesi akan dapat melewati *domain* tersebut. Pengguna harus bernegoisasi dengan *Server* yang berada dalam tingkatan di atas *domain-domain* atau penyedia layanan ini, untuk mendapatkan pengesahan (*authorization*). Terdapat peluang bagi semua *domain* untuk menentukan dimana pengguna yang sah dapat menerima permintaan layanan yang dapat diprogram. Sekali sudah ditentukan, yang mana *domain* yang akan mengambil layanan pemrograman, *domain* akan mengubah suatu kunci sesi. Kunci sesi ini digunakan oleh sistem pengguna terakhir dan suatu *programmable node* untuk melindungi *active packet*.

Solusi keamanan dalam suatu jaringan telekomunikasi

dengan banyak penyedia layanan yang dapat diprogram harus melalui beberapa fase sebagai berikut:

1. Proses pencarian dimana *domain* mengambil bagian dalam suatu sesi pada layanan terprogram.
2. Proses negosiasi untuk semua sesi dengan *domain* yang dihadapi.
3. Proses perlindungan untuk *active packet.*

Solusi pada *multidomain network* harus memenuhi kebutuhan topologi. Hal ini berarti bahwa pengguna dan *programmable node* tidak memerlukan pengetahuan topologi yang terdapat pada jaringan telekomunikasi yang dapat diprogram. Solusi yang dirancang harus terukur, dimana pemrosesan pembawaan *active packet* oleh *programmable node* tidak akan meningkat pada saat *active packet* melewati beragam *domain*. Jaringan telekomunikasi yang dapat diprogram harus berada di luar jaringan telekomunikasi, sehingga program jaringan telekomunikasi dapat selalu diambil oleh ISP yang memberikan layanan langsung ke pengguna. Oleh sebab itu, suatu sesi untuk *multidomain* akan menghasilkan dua *domain*, dan pada beberapa situasi yang ekstrim dapat meningkat jadi empat *domain.*

Terdapat penggunaan kombinasi pengarahan (*routing*), yaitu dengan pemisahan (*hashing*) dan penyaringan (*filtering*) secara konsisten.

Terdapat empat tahapan dalam arsitektur SOS, yaitu:

1. Suatu titik sumber yang merupakan bagian dari lalulintas paket informasi yang sebenarnya, meneruskan suatu paket ke suatu titik berlapis khusus, yang disebut dengan *Source Overlay Security Point (SOAP).* SOAP menerima dan meverifikasi titik sumber, sehingga suatu komunikasi

menjadi sah (*legitimate*) untuk mencapai target atau alamat yang dituju.

2. SOAP mengarahkan paket informasi ke titik simpul (*node)* khusus dalam arsitektur SOS yang memudahkannya dalam pencarian. Titik simpul khusus tersebut dinamakan dengan Beacon.
3. Beacon meneruskan paket ke suatu titik simpul rahasia (*secret node*), yang disebut dengan *secret servlet*, yang identitasnya hanya diketahui oleh sebagian kecil perangkat (*subset*) yang berada dalam jaringan telekomunikasi.
4. *Secret servlet* meneruskan paket ke alamat tujuan.
5. Penyaringan (*firewall*) akan menghentikan semua lalulintas paket informasi yang mencapai alamat tujuan tanpa melalui *secret servlet.*

Efektifitas arsitektur jaringan telekomunikasi yang diusulkan terletak pada bagaimana perangkat *client* yang sah mengetahui keberadaan *active server* dan mengetahui prosedur untuk memindahkan proses hubungan. Hal ini dapat dilakukan dengan menggunakan penyaringan tersebar (*distributed firewall*) yang dinamakan dengan arsitektur *Server Hopping*.

Untuk dapat mengetahui lokasi *active server*, suatu perangkat *client* membutuhkan dua set informasi, yaitu alamat *server* dan waktu dimana *server* aktif. Informasi ini dapat dengan mudah ditentukan dengan menggunakan serangkaian komunikasi. Untuk menghindarkan ancaman keamanan DoS di internet, perangkat *client* dan perangkat *server* memerlukan mekanisme komunikasi yang aman, yang memberikan perlindungan *privacy* dan *integrity* paket informasi. Mekanisme komunikasi tersebut adalah:

1. Mekanisme yang menentukan hubungan yang diteruskan

di antara titik-titik akhir yang dituju.

2. Mekanisme yang mengenali akibat dari lapisan jaringan telekomunikasi dan lapisan aplikasi pada semua sisi perangkat, baik *server* maupun *client*.
3. Mekanisme yang memperbaiki kondisi hubungan dan aplikasi.
4. Mekanisme perpindahan pemicu.

Dengan perlindungan SOS untuk menghadapi ancaman keamanan DoS dan DDoS, perbandingan waktu tersisa dalam beragam pengiriman paket dengan waktu pengiriman paket secara aktual.selalu tetap (konstan). Pada arsitektur *Server Hopping*, waktu pengiriman paket diperbaiki secara konstan. Dengan menambah lamanya waktu pengiriman paket secara aktual, maka waktu sisa dapat dikurangi sehingga seolah-olah tidak ada. Tentu saja hal ini akan menghasilkan kondisi dimana waktu sisa yang merupakan waktu tunda yang dapat menimbulkan penolakan layanan, dapat dikurangi atau dihilangkan.

BAB X
VIRUS, WORM, TROJAN

10.1. Virus

Virus adalah suatu program perangkat yang dapat menyebar pada perangkat atau jaringan telekomunikasi dengan cara membuat salinan dari dirinya sendiri tanpa sepengetahuan dari pengguna perangkat tersebut.

Suatu virus pertamakali harus dijalankan sebelum mampu untuk menginfeksi suatu perangkat. Berbagai macam cara agar virus ini dijalankan oleh sasaran, menempelkan dirinya pada suatu program yang lain. Terdapat juga virus yang jalan ketika dibuka suatu jenis *file* tertentu yang memanfaatkan celah keamanan jaringan telekomunikasi yang terdapat pada perangkat (baik sistem operasi atau aplikasi). Suatu *file* yang sudah terinfeksi virus dalam *attachment* e-mail. Begitu *file* tersebut dijalankan, maka kode virus akan berjalan dan mulai menginfeksi perangkat dan dapat menyebar pula ke semua *file* yang terdapat di jaringan telekomunikasi.

Virus dapat memperlambat *email* yaitu dengan membuat trafik *email* yang sangat besar yang akan membuat *server* menjadi lambat atau bahkan menjadi *crash*. Virus dapat mencuri informasi dan mampu merekam *key*stroke *keyboard*. Virus dapat menggunakan perangkat lain untuk menyerang suatu situs (*MyDoom*), merusak informasi (*Virus Compatable*), menghapus informasi (Virus Sircam), men-*disable hardware* (Virus CIH atau Chernobyl), menimbulkan hal-hal yang aneh dan mengganggu (Virus Netsky-D) dan menampilkan

informasi tertentu (Virus Cone-F).

Untuk mencegah (*prevent*) kemungkinan adanya virus sebagai salah satu ancaman atau ancaman keamanan dari luar, maka administrator jaringan telekomunikasi perlu memasang perangkat atau sistem yang mampu mendeteksi dan mencegah kemungkinan tejadinya ancaman keamanan, seperti *virus*, *worm*, trojan, atau *spyware*. Sistem pengamanan jaringan telekomunikasi yang perlu dipasang di antaranya adalah dengan pembuatan konfigurasi *Firewall* yang tepat pada seluruh perangkat jaringan telekomunikasi sebagai sistem pagar pengaman pada jaringan telekomunikasi yang dikelola. Kemudian administrator jaringan telekomunikasi perlu juga memasang *Intruder Detection System* (IDS) sebagai pemantau adanya penyusup yang mungkin berniat jahat pada jaringan telekomunikasi. Terakhir, administrator jaringan telekomunikasi wajib memasang perangkat penangkal ancaman keamanan seperti Anti Virus, Anti Worm, Anti Trojan, Anti Spyware pada jaringan telekomunikasi yang dikelolanya, dan perangkat tersebut harus selalu diperbarui setiap harinya.

Setelah menambahkan perangkat atau sistem keamanan jaringan telekomunikasi, administrator jaringan telekomunikasi wajib melakukan evaluasi atau pemeriksaan secara berkala terhadap jaringan telekomunikasi yang dikelolanya. Dengan mengevaluasi sistem dan perangkat keamanan jaringan telekomunikasi, administrator dapat mengantisipasi kemungkinan adanya sumber lubang keamanan yang baru, yang disebabkan oleh beberapa hal, seperti rancangan jaringan telekomunikasi yang kurang baik (misal: adanya *shared-net* tanpa *pemantauan*, *TCP/IP Sequence Numbering* yang mudah ditebak, *algoritma enkripsi* yang lemah), kesalahan konfigurasi

(misal: *false sense security, writeable Files for all, active default account*) dan implementasi yang kurang baik (misal: *bad programming, out of bound array, sloppy programming*). Evaluasi dapat dilakukan dengan cara manual, otomatis atau kombinasi dari keduanya.

Untuk menangani ancaman keamanan (*post attack recovery*) yang sudah terlanjur masuk dan merusak sistem adalah dengan melihat sumber ancaman keamanan dan memilih *tools* yang tepat untuk menghapus atau menghilangkannya. Selain itu, yang perlu diperhatikan adalah efek yang ditinggalkan setelah terjadinya ancaman keamanan, yang perlu untuk segera ditangani atau diperbaiki. Pada kasus *Virus Conficker* yang menyerang jaringan LAN (termasuk akses ke dalam atau intranet dan akses keluar atau internet), penanganannya dapat dilakukan dengan berbagai cara, antara lain:

Mencegah virus untuk tidak menginfeksi lebih banyak perangkat dalam satu jaringan telekomunikasi adalah dengan cara:

1. Melakukan langkah isolasi terhadap lokasi dimana virus menyerang, dengan memutuskan sambungan (*connection*) perangkat yang bermasalah terhadap jaringan telekomunikasi.
2. Mencari sumber penyebar virus di jaringan telekomunikasi. Logikanya adalah apabila semua perangkat yang belum di-*patch* dihubungkan ke jaringan telekomunikasi, dan ternyata ada satu perangkat yang terinfeksi virus Conficker, maka semua perangkat akan terinfeksi Conficker juga dalam waktu singkat, kecuali pada perangkat-perangkat yang telah dilindungi oleh

Firewall yang melindungi port: Protokol UDP pada *port* 135, 137, 138 dan 445 Protokol TCP pada *port* 135, 139, 445 dan 593.

Penanganan virus dapat dilakukan dengan cara:

1. Menggunakan *remove tools* dari Norman untuk membersihkan virus yang masih aktif.
2. Menghapus (*delete*) *service* svc*host*.exe gadungan yang ditanamkan virus pada registry.
3. Menghapus *schedule task* yang dibuat oleh virus pada C:\WINDOWS\Tasks.
4. Menghapus *string registry* yang dibuat oleh virus.
5. Menggunakan *notepad* untuk menyalin, lalu simpan dengan nama "repair.inf" (pilih *Save As Type* menjadi *All Files* agar tidak terjadi kesalahan). Jalankan repair.inf dengan klik kanan, kemudian pilih *install*.
6. Untuk *file* yang aktif pada startup dapat di-*disable* melalui "msconfig" atau dapat menghapus secara manual pada *string*: "HKLM, Software-Microsoft-Windows-CurrentVersion-Run"\
7. Untuk pembersihan virus W32/Conficker.DV secara optimal dan mencegah infeksi ulang, sebaiknya menggunakan antivirus yang terbaru dan mampu mendeteksi virus ini dengan baik.

Secara umum terdapat dua jenis program anti virus yaitu *on-access* dan *on-demand scanner*.

1. *On-access scanner* akan selalu aktif dalam sistem perangkat selama pengguna menggunakannya dan akan secara otomatis memeriksa *file-file* yang diakses dan dapat mencegah pengguna untuk menggunakan *file-file* yang sudah terinfeksi oleh virus perangkat.

2. *On-demand scanner* membiarkan pengguna yang akan memulai scanning terhadap *file-file* di perangkat. Dapat diatur penggunaannya agar dapat dilakukan secara periodik dengan menggunakan scheduler.

Beberapa perangkat lunak antivirus antara lain adalah: Avira, Norton Antivirus, McAfee VirusScan Plus, PC Tools Antivirus, Windows Live OneCare, F-Prot Antivirus, Kapersky, AVG Antivirus.

10.2. Worm

Worm dapat dikatakan mirip dengan virus tetapi worm tidak memerlukan carrier dalam hal ini program atau suatu dokumen. Worm mampu membuat salinan dari dirinya sendiri dan menggunakan jaringan telekomunikasi antar perangkat untuk menyebarkan dirinya. (*Worm Blaster*).

Banyak virus seperti MyDoom atau Bagle bekerja sebagaimana layaknya worm dan menggunakan *email* untuk mem-forward dirinya sendiri ke pihak lain. Perbedaan worm dan virus adalah Virus menginfeksi target code, tetapi worm tidak. Worm hanya menetap di memori.

10.3. Trojan

Trojan adalah program yang terlihat seperti program yang valid atau normal, tetapi sebenarnya program tersebut membawa suatu kode dengan fungsi-fungsi yang sangat berbahaya bagi perangkat. Berbeda dengan virus, *Trojan* tidak dapat memproduksi diri sendiri.

BAB XI
SPYWARE, KEYLOGGER, ADWARE, SPAM

11.1. Spyware

Spyware adalah perangkat lunak yang melacak penggunaan internet dan melaporkannya ke pihak lain, dimana proses pelacakan tidak diketahui oleh pengguna perangkat lunak tersebut. Saat ini *spyware* sudah dijadikan alat untuk mencari informasi pribadi pada suatu perangkat dan menjadikan perangkat sasaran sebagai mata-mata tanpa diketahui pengelolanya.

Ciri khas adanya *spyware* adalah:

1. Perangkat menjadi lambat, bahkan jika dijalankan tanpa menggunakan banyak program.
2. Perubahan *setting browser* dimana pengguna merasa tidak pernah mengubah atau memasangnya. Banyak kasus *start page browser* berubah tanpa sebab yang jelas dan bahkan tidak dapat diubah meskipun secara manual.
3. Gejala lain munculnya *toolbar* yang menyatu dengan komponen *toolbar browser*.
4. Kegiatan mencurigakan. Banyak pengguna melaporkan perangkat mengakses *harddisk* tanpa campur tangan pengguna. Hubungan internet menunjukkan kegiatan, meskipun pengguna tidak menggunakannya. Munculnya *icon-icon* baru yang tidak jelas pada *tray icon*. Semuanya ini menandakan adanya kegiatan *background* yang sedang bekerja pada perangkat pengguna.
5. Muncul iklan pop up setiap kali pengguna terhubungan

dengan internet. *Pop up* ini akan muncul terus-menerus meskipun sudah diclose secara manual. Isi dari *pop up* tersebut bahkan tidak terdapat hubungannya dengan situs yang sedang dibuka oleh pengguna. *Pop up* tersebut dapat berupa tampilan situs porno atau junk site lainnya.

Umumnya program jenis *spyware* masuk secara langsung dengan mengelabuhi pengguna internet. dapat saja seseorang yang membuka suatu website dan secara tidak sengaja menerima suatu peringatan dan melakukan apa yang di kehendaki oleh si pembuat web.

Spyware dapat menular lewat beberapa perangkat lunak yang di gunakan untuk pertukaran *file* video, musik dan gambar. Beberapa program yang di distribusikan bersama spyware: BearShare, Bonzi Buddy, Dope Wars, ErrorGuard, dan sebagainya.

Spyware dianggap berbahaya karena dapat:

1. Menghabiskan *resource* sistem perangkat, seperti memori dan ruang *harddisk*.
2. Mengganggu privasi pengguna dengan memberikan informasi keluar mengenai kebiasaan pengguna menggunakan perangkat.Jika suatu program spyware memasang program "*keylogger*", maka program tersebut dapat merekam pengetikan tombol *keyboard* dan mengirimkannya ke pihak lain.
3. Beberapa program *spyware* kenyataannya adalah *Trojan* yang memungkinkan seseorang masuk menjadi perangkat pengguna dan menggunakannya untuk mengirimkan *email* spam ataupun ancaman keamanan-ancaman keamanan "tak bertuan" ke perangkat lain dalam jaringan internet.

Untuk mencegah penyebaran *spyware* adalah dengan:

1. Memperhatikan apa saja *file* yang di download atau di jalankan.
2. Jangan mengunduh *file* dari sumber yang tidak jelas meliputi *web link* atau program yang dikirimkan via *email* atau messenger (YM, IM).
3. Mencari informasi tentang perangkat lunak yang akan di download atau digunakan.
4. Tidak melakukan *browsing* ke situs-situs yang berbahaya seperti situs porno, situs penyedia *cracks* atau situs lagu.
5. Tidak mengklik kata *next* pada situs tertentu. Umumnya situs yang memiliki program kutu internet mencoba mengakali pengguna internet. Cara menjebak pengguna banyak dilakukan oleh site-site porno gratis atau perangkat lunak gratis misalnya dengan memberikan warning harus berumur 17 tahun dan harus menyetujui dengan mengklik *icon*.
6. Jangan mengklik suatu *link* bila tidak yakin mendapatkan suatu *email* yang tidak jelas pengirimnya.
7. Berhati hati dengan aplikasi program yang digunakan secara gratis/*freeware*.
8. Menggunakan program *Anti Spyware* untuk dapat mencegah masuknya program yang akan mengganggu dan menyerang perangkat, di antaranya: *Spyware Doctor, Xoftspy SE Antispyware*, *Norton Internet Security, Webroot Spy Sweeper, CounterSp, Yahoo Toolbar* with Anti-Spy.

11.2. Keylogger

Pada dasarnya *keylogger* merupakan program yang berfungsi untuk mencatat hentakan *keyboard* yang dilakukan

oleh seorang pengguna ketika bekerja dengan perangkat. Kemudian fungsi ini meluas dan *keylogger* bukan hanya mampu menjalankan fungsi di atas tetapi juga mampu mendeteksi program-program atau *file* yang dijalankan. Disamping itu juga mampu mendeteksi penekanan tombol kanan dan kiri mouse. Jadilah *keylogger* suatu perangkat yang lengkap untuk memantau dan merekam kegiatan seseorang dalam menggunakan perangkat.

Keylogger dapat dibedakan menjadi dua bagian, berupa perangkat keras (*hardware*) atau perangkat lunak (*software*). *Keylogger* dalam bentuk *hardware* merupakan suatu perangkat kecil sebesar baterai berukuran AA. Perangkat ini dipasang pada ujung *keyboard* dan menjadi perantara antara *keyboard* dengan CPU. Tentunya perangkat ini akan melakukan *interception* atau pencegatan masukan data dari *keyboard*. Keunggulan perangkat ini adalah tidak terdeteksi pada tampilan *Close Program Dialog* yang tampil sewaktu pengguna menekan tombol CTRL+ALT+DEL.

Keylogger dalam bentuk *software* memerlukan instalasi. Keunggulan *keylogger* jenis ini adalah tidak menunjukkan perubahan secara fisik pada perangkat. Kelemahannya, tidak semua program *keylogger* dapat bekerja secara tersembunyi. Program ini dapat berfungsi untuk melakukan pemantauan pada seseorang untuk mencegah terjadinya penyelewengan yang dilakukan oleh seorang bawahan. Juga berfungsi untuk menguji loyalitas seseorang. Pada sisi yang lain program ini juga dapat diarahkan sebagai program untuk melakukan kegiatan yang buruk seperti pencurian data, *password*, data pribadi, dan lain-lain. Bahkan program ini dapat menjalankan peran sebagai mata-mata yang jitu! Sehingga bukan tidak

mustahil program *keylogger* ini mengambil peran cukup penting dalam dunia spionase.

Beberapa program *keylogger* berfungsi untuk melakukan *remote monitoring*. Program *keylogger* dapat mengirimkan *file log* hasil pemantauan ke suatu *email account* tertentu. Fitur ini dikenal dengan nama *email log-file delivery*. Tentunya fitur *email-log file delivery* ini akan berjalan dengan baik bila terdapat koneksi internet yang terus–menerus. Sehingga *keylogger* jenis ini dapat mengirimkan *file log* ke *email account* tertentu.

Fitur-fitur umum yang terdapat pada *keylogger*:

1. *Keystroke*
   Seluruh *keylogger* mempunyai fitur ini, yaitu, mendeteksi penekanan tombol-tombol *keyboard*.
2. *Mouse Click*
   Hanya sebagian *keylogger* saja yang mempunyai fitur ini, yaitu mampu mendeteksi penekanan tombol kiri dan kanan *mouse*.
3. *File Activity*
   Mendeteksi *file* atau program yang dijalankan oleh seorang pengguna.
4. *Log File*
   Merupakan laporan *keylogger* yang berisi catatan kegiatan pengguna.
5. Enkripsi *file log*
   File log disimpan dengan cara dienkripsi sehingga tidak dapat dibaca oleh sembarang orang.
6. *Email log-file delivery*
   Dengan fitur ini, *keylogger* akan mengirimkan *file log* ke suatu *email account* tertentu.

7. *Invisible*
   Keylogger bersifat siluman sebagaimana dijelaskan di atas.
8. *Automatically Start*
   Fitur ini memungkinkan *keylogger* untuk aktif secara otomatis ketika Windows dimulai. Program-program yang bekerja di belakang layar banyak menggunakan cara ini supaya aktif secara otomatis saat Windows dimulai seperti pada trojan, virus atau worm.
9. *Screenshot*
   Dengan fitur ini *keylogger* akan mengambil tampilan layar monitor dan menyimpannya ke dalam suatu *file* gambar. Cukup memberikan informasi tentang apa yang sedang dilakukan pengguna. Proses pengambilan gambar ini terjadi secara periodik. Inilah yang dilakukan oleh *keylogger* iOPus Starr PC dan Internet Monitor.

11.3. Adware

Adware sebenarnya difungsikan sebagai promosi atau iklan berbentuk banner yang dimasukan secara tersembunyi oleh pembuat program. Umumnya program diberikan secara gratis, tetapi dengan kompensasi pengguna harus menerima iklan pada program.

Terkadang pengguna ingin mengunakan program shareware tetapi dalamnya terdapat program yang difungsikan sebagai *Adware*. Contoh *Adware*: misalnya program yang diberikan secara gratis, ternyata memiliki jendela kecil pada program dan terus berganti ganti gambar iklan.

11.4. Spam

Spam adalah suatu *email* yang membawa informasi-

informasi yang sifatnya komersial (bisa menjual jasa, barang atau menawarkan sesuatu hal yang menarik). dapat dianalogikan sebagai suatu junk *email* yang masuk ke dalam mailbox. Spam sering kali tidak membawa informasi yang penting bagi pengelola *email* dan sangat merugikan pengguna *email.*

Spam pertamakali ditemukan pada bulan Mei 1978 dalam jaringan telekomunikasi Arpanet oleh seorang pekerja *Digital Equipment Corporation (DEC). Spammer* pertama tersebut menyalin daftar alamat *email* pada pengguna Arpanet dan mengetiknya satu persatu dalam field *carbon salinan* (CC) yang hanya mampu menampung sebanyak 320 alamat *Email*. Setelah Arpanet berkembang menjadi internet, informasi yang tergolong menjadi spam pertama dikirimkan seorang mahasiswa bernama Dave Rhodes, dengan judul *email* "Make.Money.Fast!!" dan mempostingnya di Usenet (*newsgroup*). Masih pada tahun yang sama, dua orang pengacara AS, Cantor dan Siegel mengirimkan informasi iklan "Green Card Lottery" ke 6000 newsgroup dalam waktu yang bersamaan sehingga menyeBabkan *server* Usenet *collapse* sebanyak 15 kali.

Untuk mencegah penyebaran spam adalah dengan menyaring *email* yang masuk menjadi mail box dengan mensetting konfigurasi pada *email* tersebut. Untuk outlook express, firebird, dan program *email client* yang lainnya dapat di atur dari setting/konfigurasi perangkat lunak tersebut.

BAB XII
KEAMANAN KOMUNIKASI BERBASIS WEB

12.1. World Wide Web (WWW)

World Wide Web (WWW) dikembangkan oleh Tim Berners-Lee ketika bekerja di CERN (Swiss). Untuk membaca atau melihat sistem WWW digunakan tools yang dikenal dengan istilah *browser.*

Browser awal adalah NeXT. Selain NeXT, saat itu terdapat *browser* yang berbentuk text seperti "*line mode*" *browser*. Kemudian terdapat Mosaic yang *multi-platform* (Unix/Xwindow, Mac, Windows) dikembangkan oleh Marc Andreesen dkk ketika sedang magang di NCSA.Arsitektur sistem Web terdiri dari dua sisi: s*erver* dan *client*.

Selain menyajikan informasi-informasi dalam bentuk statis, sistem Web dapat menyajikan informasi dalam bentuk dinamis dengan menjalankan program. Program ini dapat dijalankan di *server* (misal dengan CGI, servlet) dan di *client* (applet, Javascript). *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".

Pembatasan akses dapat dilakukan dengan membatasi domain atau nomor IP yang dapat mengakses; (konfigurasi Web *Server* atau *Firewall*, menggunakan pasangan *userid* dan *password*; mengenkripsi informasi sehingga hanya dapat dibuka (dekripsi) oleh orang yang memiliki kunci pembuka.

12.2. Secure Socket Layer (SSL)

Server WWW *Apache* (yang tersedia secara gratis) dapat dikonfigurasi agar memiliki fasilitas SSL dengan menambahkan perangkat lunak tambahan (SSLeay yaitu implementasi SSL dari Eric Young atau OpenSSL1 yaitu implementasi Open Source dari SSL). Penggunaan SSL memiliki permasalahan yang bergantung pada lokasi dan hukum yang berlaku. Hal ini disebabkan:

1. Pemerintah melarang ekspor teknologi enkripsi (kriptografi).
2. Paten *Public key Partners* atas *Rivest-Shamir-Adleman* (*RSA*) public*key cryptography* yang digunakan pada SSL.
3. Mengetahui Jenis *Server*
   Informasi tentang web *server* yang digunakan dapat dimanfaatkan untuk melancarkan ancaman keamanan sesuai dengan jenis *server* dan operating sistemnya.

12.3. Common Gateway Interface (CGI)

Common Gateway Interface (CGI) digunakan untuk menghubungkan sistem WWW dengan perangkat lunak lain di *server* web. Adanya CGI memungkinkan hubungan interaktif antara pengguna dan *server* web. CGI seringkali digunakan sebagai mekanisme untuk mendapatkan informasi dari pengguna melalui "*fill out form*", mengakses basis data, atau menghasilkan halaman yang dinamis.

Secara prinsip, mekanisme CGI tidak memiliki lubang keamanan jaringan telekomunikasi, program atau skrip yang dibuat sebagai CGI dapat memiliki lubang Keamanan jaringan telekomunikasi. Program CGI ini dijalankan di *server* web sehingga menggunakan *resource*s web *server* tersebut dan

membuka potensi lubang keamanan jaringan telekomunikasi.

Beberapa contoh lubang keamanan pada CGI adalah:

1. CGI dipasang oleh orang yang tidak berhak.
2. CGI dijalankan berulang-ulang untuk menghabiskan *resource*s (CPU, disk): DoS.
3. Masalah *setuid* CGI dalam sistem UNIX, dimana CGI dijalankan oleh *userid* web *server.*
4. Penyisipan karakter khusus untuk shell expansion.

Pelanggaran *privacy* (kerahasiaan) suatu sisitem, di antaranya adalah:

1. Adanya penyimpanan informasi *browsing* pada "*cookie*" yang fungsinya adalah untuk menandai kemana pengguna *browsing*.
2. Adanya situs web yang mengirimkan *script* (misal Javascript) yang melakukan interogasi terhadap *server client* (melalui *browser*) dan mengirimkan informasi ini ke *server*.

12.4. Web Deface

World Wide Web (WWW) merupakan bagian dari internet yang paling populer, sehingga ancaman keamanan paling banyak terjadi lewat *port* 80 atau yang dikenal sebagai *Web hacking*. *Web Deface* adalah tindakan untuk mengubah halaman depan atau isi suatu situs Web sehingga tampilan atau isinya sesuai dengan yang kehendaki. *Deface* banyak terjadi pada situs *e-commerce web* yang menggunakan Microsoft IIS.

Secara garis besar, *Web Deface* dapat dilakukan dengan 3 cara yaitu:

1. Memasukkan *Input Illegal*

   Tujuan adalah agar pengguna terlempar keluar dari

direktori *file-file* web *server* dan masuk ke root directory untuk kemudian menjalankan cmd.exe dan mengamati struktur direktori pada *server* sasaran.

2. Dengan TFTP (*Trivial File Transfer Protocol*) adalah protokol berbasis UDP yang listen pada *port* 69 dan sangat rawan keamanan jaringan telekomunikasinya dan kebanyakan web *server* menjalankan layanan TFTP ini.
3. Dengan FTP yang telah diisi bahan *deface*. Setiap NT *server* memiliki *file* ftp.exe untuk melakukan FTP *upload* ataupun FTP *download* (dari dan ke sever).
4. Dengan NETCAT untuk membentuk *port* penyaringan sendiri yang memungkinkan *file* transfer tanpa menggunakan FTP. Lebih lanjut, Netcat dapat digunakan untuk menghindari *port* penyaringan pada kebanyakan *firewall*, men-spoof *IP Address*, sampai melakukan *session hijacking*.

Pencegahan terhadap *Web Deface* dapat dilakukan dengan cara berikut, yaitu:

1. Mengamankan *server* IIS dari *deface*.
2. Selalu memperbarui *service* pack dan *hotfix* terbaru.
3. Melindungi dengan oleh *firewall* dan IDS.

12.5. SQL Injection

SQL Injection merupakan salah satu teknik dalam melakukan web *hacking* untuk memperoleh akses dalam sistem basis data berbasis Microsoft SQL *Server*. Teknik ini memanfaatkan kelemahan dalam bahasa pemograman scripting pada SQL dalam mengolah suatu sistem basis data yang memungkinkan seseorang tanpa *account* dapat masuk dan lolos verifikasi dari MS SQL *server*.

Bila seseorang ingin mengakses suatu sistem perangkat, biasanya diperlukan *login*. Penerapannya pada website ditampilkan dengan dua kotak isian, yaitu kotak *User Name*, *User Id*, atau *User Account* dan kotak *Password*. *Login* ini berguna untuk mem*filter* dan mengetahui identitas seseorang yang ingin mengakses suatu sistem. Sistem *login* merupakan suatu cara dalam dunia keamanan perangkat untuk mem*filter* orang yang masuk ke sistem sehingga orang-orang yang tidak terdaftar tidak dapat masuk ke sistem tersebut.

Seseorang yang tidak mengetahui *password* dapat juga ikut masuk ke suatu sistem login. Untuk menembus *password login* dapat menggunakan *SQL Injection* dengan memanfaatkan kelemahan *query* bahasa SQL. *SQL Injection* adalah suatu cara mengeksploitasi kelemahan bahasa SQL dengan memasukkan (menginjeksikan) beberapa karakter tertentu berupa *injection string*. Ada banyak hal yang dapat dilakukan dengan *injection string*. Di antaranya, mem-*bypass password*, mendapatkan nama tabel, menyisipkan anggota baru, dan lain-lain, bahkan sampai menghapus tabel pada suatu *database*.

BAB XIII
KEAMANAN JARINGAN LOKAL NIRKABEL

13.1. Pengenalan Jaringan Lokal Nirkabel (WLAN)

Jaringan Lokal Nirkabel/*Wireless Local Area Network* (WLAN) adalah suatu teknologi yang memungkinkan pengiriman informasi dengan kecepatan antara 11-54 *Megabyte persecond*. Teknologi ini dikenal dengan sebutan *Wireless Fidelity (Wi-Fi)* yang membuat pengguna internet berkomunikasi informasi secara nirkabel. WLAN sebenarnya hampir sama dengan jaringan LAN, akan tetapi setiap *node* pada WLAN menggunakan *wireless device* untuk berhubungan dengan jaringan telekomunikasi. *Node* pada WLAN menggunakan *channel* frekuensi yang sama dan SSID yang menunjukkan identitas *wireless device*.

WLAN memiliki dua mode yang dapat digunakan, yaitu: Infastruktur dan Ad-Hoc. Konfigurasi Infrastruktur adalah komunikasi antara masing-masing PC melalui suatu *Access Point* pada WLAN atau LAN. Komunikasi *Ad-Hoc* adalah komunikasi secara langsung antara masing-masing perangkat dengan menggunakan kartu antarmuka jaringan nirkabel (*wireless network interface card).*

Umumnya, komponen WLAN terdiri atas:

1. *Access Point*, berfungsi untuk mengkonversikan sinyal frekuensi radio (RF) menjadi sinyal digital yang akan disalurkan melalui kabel, atau disalurkan ke perangkat WLAN yang lain dengan dikonversi ulang menjadi sinyal frekuensi radio.

2. *Mobile/Desktop PC*, merupakan perangkat akses untuk pengguna yang umumnya sudah terpasang *port* PCMCIA sedangkan desktop PC harus ditambahkan wireless adapter melalui *Peripheral Component Interconnect* (PCI) card atau *Universal Serial Bus* (USB).
3. *Antena external (optional)* digunakan untuk memperkuat daya pancar. Antena ini dapat dirakit sendiri oleh pengguna, misalnya: antena kaleng.

13.2. Perlindungan WEP dan WPA

Komponen *logic* dari *Access Point* adalah *Extended Service Set Identification* (ESSID) yang merupakan standar dari IEEE 802.11. Pengguna harus menghubungkan *wireless adapter* ke *Access Point* dengan ESSID tertentu supaya transfer informasi dapat terjadi. ESSID menjadi otentifikasi standar dalam komunikasi wireless. Dalam segi keamanan jaringan telekomunikasi, beberapa *vendor* tertentu membuat kunci otentifikasi tertentu untuk proses otentifikasi dari *client* ke *Access Point*.

Rawannya segi keamanan jaringan telekomunikasi nirkabel membuat IEEE mengeluarkan standarisasi *Wireless Encryption Protocol* (WEP), yaitu suatu aplikasi yang sudah terdapat dalam setiap *PCMCIA card*. WEP berfungsi mengenkripsi informasi sebelum ditransfer ke sinyal *Radio Frequency* (RF), dan mendekripsi kembali informasi dari sinyal RF.

WEP disebut juga *Shared Key*, yaitu metode otentikasi yang membutuhkan enkripsi-dekripsi. Enkripsi WEP menggunakan kunci yang dimasukkan (oleh administrator) ke *client* maupun *Access Point*. Kunci ini harus cocok dari yang

diberikan *Access Point* ke *client*, dengan yang dimasukkan *client* untuk otentikasi ke *Access Point*.

Wi-Fi Protected Access (WPA) adalah suatu sistem yang juga dapat diterapkan untuk mengamankan jaringan telekomunikasi nirkabel. Metoda pengamanan dengan WPA ini dibuat untuk melengkapi WEP dimana WEP memiliki banyak celah dan kelemahan keamanan jaringan telekomunikasi nirkabel. WPA menerapkan standar IEEE 802.11i. WPA dirancang dan digunakan dengan perangkat yang berfungsi sebagai *authentication server*, yang memberikan *key* yang berbeda kepada masing–masing pengguna/*client* pada suatu jaringan telekomunikasi nirkabel yang menggunakan *Access Point* sebagai media pusat komunikasi. Sebagaimana WEP, metoda enkripsi WPA ini juga menggunakan algoritma RC4. Pengamanan jaringan telekomunikasi nirkabel dengan metoda WPA dapat ditandai dengan minimal tiga pilihan yang harus diisi administrator jaringan telekomunikasi agar jaringan telekomunikasi dapat beroperasi pada mode WPA, yaitu:

1. *Authentication Server* adalah perangkat server yang dituju oleh *Access Point* untuk memberi otentikasi kepada *client* dengan menggunakan perangkat lunak, seperti: freeRADIUS, openRADIUS dan lain-lain.
2. Nomor port, dimana nomor port yang digunakan adalah 1812.
3. *Shared Secret* adalah kunci yang dibagikan oleh *Authentication Server* kepada *client* secara transparan.

Wi-Fi Protected Access-Pre Shared Key (WPA-PSK) adalah pengamanan jaringan telekomunikasi nirkabel yang tidak menggunakan *authentication server*. Dengan demikian, *Access Point* dapat dijalankan dengan mode WPA tanpa

menggunakan bantuan perangkat server. Perlu diketahui bahwa tidak semua *Access Point* mempunyai fasilitas yang sama dan tidak semua *Access Point* menggunakan cara yang sama dalam mendapatkan *Shared-Key* yang akan dibagikan ke *client*. Pada *Access Point* D*link* DWL-2000AP, pemberian *Shared-Key* dilakukan secara manual tanpa mengetahui algoritma apa yang digunakan. Keadaan ini berbanding terbalik dengan *Access Point Link*sys WRT54G, dimana administrator dapat memilih dua algoritma WPA yang disediakan, yang terdiri dari algoritma TKIP atau algoritma AES. Setelah *Shared-Key* didapat, maka *client* yang akan bergabung dengan *Access Point* cukup memasukkan angka/kode yang diijinkan dan dikenal oleh *Access Point*. Prinsip kerja yang digunakan WPA-PSK sangat mirip dengan pengamanan jaringan telekomunikasi nirkabel WEP dengan menggunakan metoda *Shared-Key*.

Algoritma *Temporal Key Integrity Protocol* (TKIP) adalah suatu algoritma yang didefinisikan oleh IEEE 802.11i dalam pengamanan WPA-PSK. TKIP menggunakan skema kunci berdasarkan RC4 dengan mengenkripsi semua paket informasi yang dikirimkan dengan kunci enkripsi yang unik. Setelah *Shared-Key* didapat, maka *client* yang akan bergabung dengan *Access Point* cukup memasukkan angka/kode yang diijinkan dan dikenal oleh *Access Point*. Selain itu, pengamanan WPA-PSK juga dapat menggunakan *Shared-Key. Advanced Encryption Standard* (AES) yang diperkenalkan pada bulan Desember 2001 merupakan algoritma yang lebih efisien dibandingkan algoritma sebelumnya dengan kunci sepanjang 128 bit, 192 dan 256 bit.

WPA2 adalah WPA yang diuji dan disertifikasi oleh Wi-Fi Alliance. WPA2 mendukung CCMP, mode enkripsi berbasis

AES. WPA2 dimulai pada bulan September 2004 dan mulai diwajibkan sejak 13 Maret 2006 untuk semua perangkat yang memiliki logo Wi-Fi.

Peningkatan WPA2 yang paling penting atas WPA adalah penggunaan algoritma *Advanced Encryption Standard (AES*). AES mengenkripsi informasi yang diklasifikasikan sebagai rahasia, sehingga harus cukup baik untuk melindungi jaringan telekomunikasi. Saat ini, kerentanan utama sistem WPA2 adalah ketika penyerang sudah memiliki akses ke jaringan WiFi yang aman dan dapat memperoleh akses ke kunci tertentu untuk melakukan serangan pada perangkat lain di jaringan telekomunikasi. Selain itu, kemungkinan serangan melalui *Wi-Fi Protected Setup (WPS)* masih tinggi dalam jalur akses yang memiliki akses WPA2 saat ini yang juga merupakan masalah dengan WPA.

Di bulan Januari 2018, Wi-Fi Alliance mengumumkan WPA3 sebagai pengganti WPA2. Standar baru ini menggunakan enkripsi 128 bit. Standar WPA3 juga menggantikan *Pre Shared Key* sebelumnya dengan Simultaneous Authentication of Equals sehingga menghasilkan pertukaran kunci awal yang lebih aman dalam mode pribadi. Wi-Fi Alliance juga mengklaim bahwa WPA3 akan mengurangi masalah keamanan yang ditimbulkan oleh sandi yang lemah dan menyederhanakan proses pengaturan perangkat tanpa antarmuka tampilan.

13.3. Scanning Tools

Sebagian besar jaringan WLAN diamankan dengan fitur keamanan seperti enkripsi keamanan (WEP/WPA), penyaringan alamat MAC, menonaktifkan DHCP dan

menggunakan IP statis, serta menyembunyikan SSID untuk membantu menjaga jaringan telekomunikasi nirkabel atau WIFI agar tetap aman.

13.4. Sniffing Tools Dalam WLAN

Secara umum, WLAN dikenal dengan protokol IEEE 802.11. Saat ini ada 5 spesifikasi berdasarkan IEEE 802.11 yaitu: 802.11, 802.11a, 802.11b, 802.11g dan 802.11n. Masing-masing memiliki perbedaan fitur dengan tujuan yang berbeda.

Wireshark adalah piranti lunak yang spesifik untuk melakukan analisis paket data pada jaringan telekomunikasi secara *real time* serta menghadirkan hasil analisis paket data dalam format yang dipahami oleh pengguna. *Wireshark* dapat melakukan *packet filtering*, *packet color coding*, serta beberapa fitur lain yang dapat mengizinkan untuk melihat *detil network traffic* serta pemeriksaan paket data secara individu.

Wireshark mempunyai fitur termasuk *display filter language* yang banyak dan kemampuannya dalam satu aliran pada sesi TCP. Sebagai p*acket sniffer*, *Wireshark* merupakan *tool* yang berkemampuan menahan dan melakukan pencatatan pada trafik data dalam jaringan telekomunikasi. Pada saat data mengalir dalam jaringan telekomunikasi, *packet sniffer* dapat menangkap *protocol data unit* (PDU), melakukan *decoding* juga analisis isi paket. *Wireshark* dapat melakukan *troubleshoot* persoalan jaringan telekomunikasi, pengujian permasalahan keamanan, *debugging* implementasi protokol, belajar protokol. *Wireshark* ini dapat disebut sebagai *tools* yang powerfull karena dapat digunakan untuk mencuri informasi yang sensitif pada jaringan telekomunikasi, seperti *password*, *cookie,* dan lain sebagainya.

BAB XIV
KEAMANAN WIDE AREA NETWORK (WAN)

14.1. DMZ

Pembagian kelompok jaringan telekomunikasi perlu dilakukan untuk mengatasi kemungkinan terjadinya gangguan keamanan pada suatu kelompok jaringan telekomunikasi. Suatu jaringan telekomunikasi dapat dibedakan antara jaringan lokal (*local area network*) dan jaringan luar (jaringan telekomunikasi pihak luar) dengan menggunakan *Demilitarized Zone (DMZ*). Perangkat-perangkat DMZ adalah perangkat-perangkat yang perlu dihubungi secara langsung oleh pihak luar. Contohnya adalah *web-server*, *mail-exchange*, *server* dan *name server*. Perangkat-perangkat pada DMZ perlu disiapkan secara khusus karena terbuka ke luar (dapat diakses dari pihak luar). Aplikasi-aplikasi yang digunakan pada *host-host* dalam DMZ harus merupakan aplikasi-aplikasi yang aman, terus menerus dipantau dan diperbarui secara periodic.

Aturan-aturan yang berlaku dalam DMZ adalah:

1. Pihak luar hanya dapat berhubungan dengan *host-host* dalam DMZ sesuai dengan kebutuhan yang ada. Secara *default*, pihak luar tidak dapat melakukan hubungan dengan *host-host* di jaringan lokal setelah DMZ.
2. *Host-host* pada DMZ secara *default* tidak dapat melakukan hubungan dengan *host-host* pada jaringan lokal, hubungan secara terbatas dapat dilakukan sesuai dengan kebutuhan.
3. *Host-host* pada jaringan lokal dapat melakukan hubungan secara bebas baik ke *host* maupun ke DMZ. Pada beberapa

penerapan, untuk meningkatkan keamanan, *host-host* pada jaringan lokal tidak dapat melakukan hubungan ke *host*, melainkan melalui perantara *host* (*proxy*) pada DMZ, sehingga pihak luar tidak mengetahui keberadaan *host-host* pada jaringan lokal.

Selain meningkatkan keamanan, pembagian dengan DMZ dapat menghemat penggunaan alamat IP, karena hanya *host-host* yang berada pada DMZ saja yang perlu menggunakan alamat *IP Public*, sedangkan *host-host* yang berada pada jaringan lokal dapat menggunakan alamat *IP Private*. Dalam suatu organisasi atau perusahaan, adanya pembagian kelompok jaringan telekomunikasi memerlukan adanya panduan interaksi dengan jarinagn yang dilakukan dan dibutuhkan oleh setiap bagian organisasi atau perusahaan tersebut. Aturan dasar yang banyak digunakan adalah menutup semua lubang (*port*) yang ada dan hanya membuka yang dibutuhkan dan yang aman saja. Semakin banyak pembagian kelompok jaringan telekomunikasi yang ada, maka semakin meningkatkan kompleksitas pemeliharaan jaringan telekomunikasi. Selain itu, semakin banyak pembagian kelompok juga akan meningkatkan latensi koneksi antara satu *host* di suatu kelompok jaringan telekomunikasi dengan *host* lain di kelompok jaringan telekomunikasi lainnya.

14.2. Keamanan VPN

*Virtual Private Network (VPN) b*erkembang pada saat perusahaan besar memperluas jaringan komunikasi bisnisnya, namun tetap dapat menghubungkan jaringan lokal (*private*) antar kantor cabang dengan perusahaan mitra kerjanya yang berada di tempat yang jauh. Perusahaan juga ingin memberikan

fasilitas ke karyawannya (yang memiliki hak akses) yang ingin terhubung ke jaringan lokal milik perusahaan dimanapun berada. Perusahaan tersebut perlu suatu jaringan lokal yang jangkauannya luas, tidak dapat diakses oleh sembarang orang, tetapi hanya orang yang memiliki hak akses saja yang dapat terhubung ke jaringan lokal tersebut.

VPN merupakan suatu jaringan pribadi yang menghubungkan satu *node* jaringan telekomunikasi ke *node* jaringan telekomunikasi lainnya dengan menggunakan jaringan internet. Data yang dilewatkan akan diencapsulation (dibungkus) dan dienkripsi, supaya data tersebut terjamin kerahasiaannya. Peningkatan penggunaan koneksi VPN dari tahun ke tahun karena murahnya infrastruktur yang dibutuhkan oleh VPN serta mudahnya dalam instalasi, maka koneksi ini lebih efisien dibandingkan dengan metode WAN. Jaringan VPN dikoneksikan oleh ISP lewat *router*nya ke *router-router* lain dengan mengunakan jalur internet yang telah dienkripsi antara dua titik, dengan menggunakan *leased line* untuk hubungan jarak jauh dengan VPN, perusahaan dapat menghemat 20 sampai 40% biaya WAN. Ada beberapa alasan mengapa saat ini penggunaan perusahaan banyak membangun solusi VPN, seperti:

1. Menekan biaya interkoneksi.
2. Memperluas interkoneksi ke pengguna yang selama ini susah dijangkau.
3. Dapat mengirimkan aplikasi-aplikasi baru berbasis IP.
4. Fleksibel dalam pemilihan topologi.
5. Meningkatkan *scalability network* dan tingkat keamanan.

VPN dianggap aman karena VPN menggunakan

sistem keamanan yang berlapis, diantaranya ;

1. Metode *tunneling* (terowongan), membuat terowongan *virtual* di atas jaringan komunikasi publik menggunakan protocol seperti *Point to Point Protocol* (PPTP), *Layer 2 Tunneling Protocol* (L2TP), *Generic Routing Encapsulation* (GRE) dan IPSec. PPTP dan L2TP adalah layer 2 *tunneling protocol*. Keduanya melakukan pembungkusan *payload* pada *frame Point to Point Protocol* (PPP) untuk di lewatkan pada jaringan komunikasi. IPSec berada di layer 3 yang menggunakan packet, yang akan melakukan pembungkusan IP header sebelum dikirim ke jaringan komunikasi.
2. Metode Enkripsi untuk *Encapsulations* (pembungkusan) paket data yang lewat dalam tunneling, data yang dilewatkan pada pembungkusan tersebut, data disini akan diubah dengan metode algoritma kripthography tertentu seperti DES, 3DES, atau AES.
3. Metode Otentikasi Pengguna, karena banyak pengguna yang akan mengakses biasanya digunakan beberapa metode otentikasi pengguna seperti *Remote* Access Dial In *User Service*s (RADIUS) dan *Digital Certificates*.
4. Integritas Data, paket data yang dilewatkan di jaringan komunikasi publik perlu penjaminan integritas data/kepercayaan data apakah terjadi perubahan atau tidak. Metode VPN menggunakan HMA C-MD5 atau HMA C-SHA1 untuk menjadi paket tidak diubah pada saat pengiriman.

Implementasi jaringan komunikasi dapat dilakukan dengan menggunakan *leased line*. Namun biaya yang

dibutuhkan untuk membangun infrastuktur jaringan komunikasi yang luas menggunakan *leased line* sangat besar. Di sisi lain perusahaan ingin mengoptimalkan biaya untuk membangun jaringan komunikasi yang luas. Oleh karena itu VPN dapat digunakan sebagai teknologi alternatif untuk menghubungkan jaringan lokal yang luas dengan biaya yang relatif kecil, karena transmisi data teknologi VPN menggunakan media jaringan komunikasi publik.

VPN merupakan suatu jaringan lokal yang terhubung melalui media jaringan komunikasi publik. Infrastruktur publik yang paling banyak digunakan adalah internet. Untuk memperoleh komunikasi yang aman (*private*) melalui internet, diperlukan protokol khusus untuk mengatur pengamanan datanya. Perusahaan/organisasi yang ingin membuat *wide area network* (WAN) dapat menggunakan VPN sebagai alternatif dalam implementasinya. Penggunaan *leased line* sebagai implementasi WAN membutuhkan investasi yang sangat besar.

Ada beberapa keuntungan yang dapat diperoleh dengan menggunakan VPN untuk implementasi WAN. Pertama, jangkauan jaringan lokal yang dimiliki suatu perusahaan akan menjadi luas, sehingga perusahaan dapat mengembangkan bisnisnya di daerah lain. Waktu yang dibutuhkan untuk menghubungkan jaringan lokal ke tempat lain juga semakin cepat, karena proses instalasi infrastruktur jaringan komunikasi dilakukan dari perusahaan/kantor cabang yang baru dengan ISP terdekat di daerahnya. Sedangkan penggunaan *leased line* sebagai WAN akan membutuhkan waktu yang lama untuk membangun jalur koneksi khusus dari kantor cabang yang baru dengan perusahaan induknya. Dengan demikian penggunaan VPN secara tidak langsung akan meningkatkan efektivitas dan

efisiensi kerja.

Kedua, penggunaaan VPN dapat mengurangi biaya operasional bila dibandingkan dengan penggunaan *leased line* sebagai cara tradisional untuk mengimplementasikan WAN. VPN dapat mengurangi biaya pembuatan jaringan komunikasi karena tidak membutuhkan kabel (*leased line*) yang panjang. Penggunaan kabel yang panjang akan membutuhkan biaya produksi yang sangat besar. Semakin jauh jarak yang diinginkan, semakin meningkat pula biaya produksinya. VPN menggunakan internet sebagai media komunikasinya. Perusahaan hanya membutuhkan kabel dalam jumlah yang relatif kecil untuk menghubungkan perusahaan tersebut dengan pihak ISP (*Internet Service Provider*) terdekat.

Ketiga, penggunaan VPN akan meningkatkan skalabilitas. Perusahaan yang tumbuh pesat akan membutuhkan kantor cabang baru di beberapa tempat yang terhubung dengan jaringan lokal kantor pusat. Bila menggunakan *leased line*, penambahan satu kantor cabang membutuhkan satu jalur untuk membangun WAN. Penambahan satu kantor cabang baru lagi (dua kantor cabang) akan membutuhkan dua tambahan jalur, masing-masing ke kantor pusat dan ke kantor cabang terdahulu.

Berbeda dengan penggunaan *leased line*, penambahan satu kantor cabang hanya membutuhkan satu jalur, yaitu jalur yang menhubungkan kantor cabang yang baru dengan ISP terdekat. Selanjutnya jalur dari ISP akan terhubung ke internet yang merupakan jaringan komunikasi global. Dengan demikian penggunaan VPN untuk implementasi WAN akan menyederhanakan topologi jaringan komunikasinya.

Keempat, VPN memberi kemudahan untuk diakses dari

mana saja, karena VPN terhubung ke internet. Sehingga karyawan yang mobile dapat mengakses jaringan komunikasi khusus perusahaan dimanapun dia berada. Selama dia dapat mendapatkan akses ke internet ke ISP terdekat, karyawan tersebut tetap dapat melakukan koneksi dengan jaringan komunikasi khusus perusahaan. Hal ini tidak dapat dilakukan jika menggunakan *leased line* yang hanya dapat diakses pada terminal tertentu saja.

VPN juga memiliki kelemahan yaitu, pertama, VPN membutuhkan perhatian yang serius pada keamanan jaringan komunikasi publik (internet). Oleh karena itu diperlukan tindakan yang tepat untuk mencegah terjadinya hal-hal yang tidak diinginkan seperti penyadapan, hacking dan tindakan cyber crime pada jaringan VPN.

Kedua, ketersediaan dan performansi jaringan komunikasi khusus perusahaan melalui media internet sangat tergantung pada faktor-faktor yang berada di luar kendali pihak perusahaan. Kecepatan dan keandalan transmisi data melalui internet yang digunakan sebagai media komunikasi jaringan VPN tidak dapat diatur oleh pihak pengguna jaringan VPN, karena trafik yang terjadi di internet melibatkan semua pihak pengguna internet di seluruh dunia.

Ketiga, perangkat pembangun teknologi jaringan VPN dari beberapa *vendor* yang berbeda ada kemungkinan tidak dapat digunakan secara bersama-sama karena standar teknologi VPN belum memadai. Oleh karena itu fleksibilitas dalam memilih perangkat yang sesuai dengan kebutuhan dan keuangan perusahaan sangat kurang.

Keempat, VPN harus mampu menampung protokol lain selain IP dan teknologi jaringan lokal yang sudah ada. Akan

teteapi IP masih dapat digunakan VPN melalui pengembangan IPSec (IP Security Protocol).

Beberapa metode pengamanan data yang dapat dilakukan pada teknologi jaringan VPN antara lain dengan menggunakan *firewall*. Pengamanan dapat juga dilakukan dengan melakukan enkripsi pada data yang akan dikirim melalui intenet. Selain itu, data dapat juga dikirim menggunakan protokol khusus yang aman untuk tranmisi data melalui internet (IPSec). Alternatif lain pengendalian keamanan jaringan VPN adalah dengan menggunakan metode AAA *server* yang akan memeriksa autentikasi, autorisasi dan merekam segala sesuatu yang dilakukan pengguna pada suatu jaringan komunikasi.

DAFTAR PUSTAKA

[1] C. Paar and J. Pelzl, Understanding Cryptography: A Textbook for Students and Practitioners. Springer Berlin Heidelberg, 2011, isbn: 9783642041006.[Online]. https://books.google.com.eg/books?id=N_e4NAEACAAJ

[2] Budi Rahardjo, "Keamanan Sistem Informasi Berbasis Internet", Penerbit. PT. Insan Indonesia, Bandung, 2005.

[3] Richardus Eko Indrajit, "Peranan Teknologi Informasi dan Internet", Penerbit. Andi Offset, Yogyakarta, 2011.

[4] Sri Hartanto, "Pencegahan dan Pendeteksian Serangan Penolakan Layanan (Denial of Service Attack) Dalam Jaringan komunikasi", Jurnal Ilmiah Elektrokrisna, Vol. 1, No. 3, pp.133-144, 2013.

[5] Luo, Xiapu., W.W.Chan, Edmond., and K.C. Chang, Rocky, "Detecting Pulsing Denial-of-Service Attacks with Nondeterministic Attack Intervals", EURASIP Journal on Advances in Signal Processing, Vol. 2009, No. 1, pp. 1-13, 2009, DOI:10.1155/2009/256821.

[6] Meenakshi, S and S.K Srivatsa, "A Distributed Framework with less False Positive Ratio Against Distributed Denial of Service Attack", Information Technology Journal, Vol. 6, No. 8, pp. 1139-1145, 2007, DOI:10.3923/itj.2007.1139.1145.

[7] S. H. C. Haris, "Anomaly Detection of IP Header Threats", International Journal of Advanced Computer Science and Applications, Vol. 4, No. 5, pp. 497–504, 2011.

BUKU AJAR TEKNIK ELEKTRO

KEAMANAN DAN KEHANDALAN JARINGAN

SRI HARTANTO

Buku Ajar Keamanan dan Kehandalan Jaringan ini disusun sebagai bahan pembelajaran bagi mahasiswa yang mengikuti perkuliahan Keamanan dan Kehandalan Jaringan. Buku Ajar Keamanan dan Kehandalan Jaringan ini berisikan pengenalan keamanan dan kehandalan jaringan, seperti kriptografi dan steganografi, kunci enkripsi dan dekripsi, evaluasi keamanan jaringan, keamanan jaringan berbasis sistem server, ancaman terhadap keamanan jaringan, keamanan jaringan berbasis protokol komunikasi, firewall, IDS dan IPS, virus, worm, trojan, spyware, keylogger, adware, spam, keamanan jaringan berbasis web, keamanan jaringan lokal nirkabel dan keamanan jaringan wide area network. Buku Ajar Keamanan dan Kehandalan Jaringan bertujuan untuk memperkaya pemahaman tentang pentingnya keamanan suatu jaringan telekomunikasi, dan mengenali potensi atau bahaya yang mengancam jaringan telekomunikasi.

ISBN: 978-623-145-089-0

9 786231 450890

PENERBIT CV. MITRA ILMU

IKAPI