BAB I
PENDAHULUAN

Pada bab ini akan dijelaskan mengenai beberapa hal dasar dalam Tugas Akhir ini yang meliputi latar belakang, perumusan masalah, batasan, tujuan dan manfaat pembuatan Tugas Akhir serta metodologi dan sistematika pembuatan buku Tugas Akhir ini. Dari uraian dibawah ini diharapkan gambaran Tugas Akhir secara umum dapat dipahami dengan baik.

1.1 Latar Belakang

Semakin pesatnya perkembangan teknologi informasi memudahkan orang-orang untuk saling tukar menukar data baik melalui internet maupun intranet. Tentunya dengan mudahnya berbagi data itulah sangat memungkinkan terjadinya serangan terhadap data tersebut terutama melalui jaringan komputer. Sistem pendeteksi intrusi atau yang pada umumnya disebut IDS (*Intrusion Detection System*) merupakan senjata utama untuk mengamankan suatu jaringan dimana sistem ini nantinya bertugas untuk mengidentifikasi dan mencatat apakah suatu paket data tersebut merupakan bentuk serangan atau paket data bisa.

Saat ini telah banyak dikembangkan aplikasi IDS (*Intrusion Detection System*), namun sebagian besar yang dikembangkan berbasis *signature* atau menggunakan *rule*, dan sebagian kecil menggunakan *anomaly*. *Anomaly* pada dasarnya adalah mencari data yang menyimpang dari sekumpulan data normal. IDS yang berbasis pada *anomaly* bersifat lebih fleksibel, karena dapat mengenali pola serangan baru tanpa harus memperbaharui basis data pola serangan. IDS yang berbasis pada anomali memiliki sebuah kecerdasan buatan yang mampu mendeteksi dan mengenali sebuah serangan. IDS yang berbasis anomali menggabungkan metode analisis dan statistik untuk mengenali penyimpangan tersebut. Kelemahan dari metode ini adalah kemungkinan salah identifikasi pada data yang diolah.

Sistem kerja intrusi ini pada dasarnya dikirimkan lewat jaringan dengan paket-paket data yang sama dengan paket data normal. Dengan banyaknya paket data yang masuk kedalam sebuah host, tentunya host ini harus bisa mengenali paket data, apakah paket data tersbut terdapat paket data yang berupa intrusi atau tidak. Hal tersebut dapat dikenali dengan cara mengelompokkan data berdasarkan beberapa hal yang membedakan antara paket data normal dengan paket data yang berupa intrusi.

Maka untuk membedakan hal tersebut diperlukan sebuah sistem deteksi intrusi dimana nantinya sistem deteksi intrusi tersebut menggunakan gabungan metode analisis dan statistik yang berfungsi mengenali perbedaan paket data normal maupun paket data berupa intrusi. Selain itu, sistem deteksi intrusi yang dapat mempelajari paket data normal yang baru sebagai data *training*.

Untuk dapat menghitung jarak mahalanobis dari paket data, diperlukan metode yang dapat merubah informasi paket data menjadi nilai yang dapat dihitung. Metode n-gram dapat digunakan untuk mebuat model yang sederhana dan cepat untuk dihitung khususnya menghitung distribusi karakter pada suatu paket data. N-Gram merupakan metode yang paling efesien dan efektif dalam membuat model dari suatu paket data.

Metode mahalanobis *distance* berguna untuk membedakan paket-paket data berdasarkan anomali yang terjadi. Untuk dapat mempelajari paket data normal yang baru menggunakan metode *incremental learning*, dimana metode ini nantinya memperbaharui rata-rata dan standar deviasi dari model paket data yang ada pada data *training*.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana membangun sistem deteksi intrusi yang dapat membaca data set dari DARPA IDS tahun 1999 data set?

2. Bagaimana membangun sistem deteksi intrusi yang dapat menangkap paket data dari *network interface* suatu komputer?
3. Bagaimana menerapkan metode n-gram pada konten paket data?
4. Bagaimana cara mengklasifikasikan paket data menjadi dua kelompok, yaitu paket data normal dan paket data yang berupa intrusi dengan menggunakan metode Mahalanobis distance?
5. Bagaimana membangun sistem deteksi intrusi yang menerapkan metode *incremental learning*?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, yaitu sebagai berikut:

1. Data set yang digunakan adalah DARPA IDS tahun 1999.
2. Jenis protokol yang akan diperiksa adalah TCP dengan port aplikasi dari FTP(21), Telnet(23), SMTP(25), HTTP(80) dan UDP dengan port aplikasi dari DNS Server(53).

1.4 Tujuan

Tujuan dari dibuatnya Tugas Akhir ini adalah membuat sistem pendeteksi intrusi yang mampu mengenali serangan pada lalu lintas jaringan dengan menggunakan metode Mahalanobis Distance berbasis anomali yang nantinya mampu membedakan paket data normal maupun paket data yang berupa intrusi.

1.5 Manfaat

Dengan dibuatnya Tugas Akhir ini akan memberikan konsep baru pada cara membedakan paket data normal dan paket data yang berupa intrusi.

1.6 Metodologi

Tahapan-tahapan yang dilakukan dalam pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. Penyusunan proposal Tugas Akhir.
   Tahap awal untuk memulai pengerjaan Tugas Akhir adalah penyusunan proposal Tugas Akhir. Proposal Tugas Akhir yang diajukan memiliki gagasan yang sama dengan Tugas Akhir ini, yaitu membuat aplikasi deteksi intrusi pada jaringan komputer berbasi anomali dengan *n-gram* dan *incremental learning*.

2. Studi literatur
   Pada tahap ini dilakukan pemahaman informasi dan literatur yang diperlukan untuk pembuatan implementasi program. Tahap ini diperlukan untuk membantu memahami pengunaan komponen-komponen terkait dengan sistem yang akan dibangun, antara lain : IDS secara umum, IDS berbasis anomali, metode Mahalanobis *Distance*, metode *n-gram*, metode *incremental learning* dan Jpcap.

3. Analisis dan desain perangkat lunak
   Tahap ini meliputi perancangan sistem berdasarkan studi literatur dan pembelajaran konsep teknologi dari perangkat lunak yang ada. Tahap ini mendefinisikan alur dari implementasi. Langkah-langkah yang dikerjakan juga didefinisikan pada tahap ini. Pada tahapan ini dibuat *prototype* sistem, yang merupakan rancangan dasar dari sistem yang akan dibuat. Serta dilakukan desain fungsi yang akan dibuat yang ditunjukkan melalui diagram alir. Fungsi utama yang akan dibuat pada tugas akhir ini meliputi fungsi *sniffer*, rekonstruksi paket data, menghitung *n-gram* paket data, menghitung jarak mahalanobis dan *incremental learning*.

4. Implementasi perangkat lunak
   Implementasi merupakan tahap membangun rancangan program yang telah dibuat. Pada tahapan ini merealisasikan apa yang terdapat pada tahapan sebelumnya, sehingga

menjadi sebuah program yang sesuai dengan apa yang telah direncanakan.

5. Pengujian dan evaluasi
   Pada tahapan ini dilakukan uji coba pada data yang telah dikumpulkan. Tahapan ini dimaksudkan untuk mengevaluasi kesesuaian data dan program serta mencari masalah yang mungkin timbul dan mengadakan perbaikan jika terdapat kesalahan pada program.

6. Penyusunan buku Tugas Akhir.
   Pada tahapan ini disusun buku yang memuat dokumentasi mengenai pembuatan serta hasil dari implementasi perangkat lunak yang telah dibuat.

1.7 Sistematika Penulisan Laporan Tugas Akhir

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir secara keseluruhan. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini:

Bab I Pendahuluan
Bab yang berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan Tugas Akhir. Selain itu permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

Bab II Tinjauan Pustaka
Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan Tugas Akhir ini. Dasar teori yang digunakan adalah IDS secara umum, IDS berbasis anomali, metode Mahalanobis *Distance*, metode *n-gram*, metode *incremental learning* dan Jpcap.

Bab III Desain dan Perancangan

Bab ini berisi tentang rancangan sistem yang akan dibangun dan disajikan dalam bentuk diagram alir. Fungsi utama yang akan dibuat pada tugas akhir ini meliputi fungsi *sniffer,* rekonstruksi paket data, menghitung *n-gram* paket data, menghitung jarak mahalanobis dan *incremental learning*.

Bab IV Implementasi

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa kode program yang digunakan untuk proses implementasi.

Bab V Uji Coba Dan Evaluasi

Bab ini menjelaskan kemampuan perangkat lunak dengan melakukan pengujian kebenaran dan pengujian kinerja dari sistem yang telah dibuat.

Bab VI Kesimpulan Dan Saran

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.

BAB II
TINJAUAN PUSTAKA

Pada bab ini akan dibahas mengenai teori yang menjadi dasar dari pembuatan Tugas Akhir ini. Teori yang dibahas mencakup elemen-elemen yang terkait dalam topik Tugas Akhir mulai dari sumber dari permasalahan, pendekatan yang digunakan, serta metode dan teknologi yang digunakan untuk pengerjaan Tugas Akhir ini.

2.1 IDS

IDS (*Intrusion Detection System*) [1] adalah aplikasi perangkat lunak yang digunakan untuk menyiapakan tindakan untuk menghadapi sebuah serangan. Hal ini dijalankan dengan cara mengumpulkan informasi dari berbagai sumber, baik dari suatu sistem maupun jaringan, lalu menganalisanya untuk menentukan ada tidaknya ancaman. IDS dikategorikan menjadi 3, yaitu [1]:

a. NIDS (*Network Intrusion Detection System*), menjalankan analisa terhadap *traffic* dalam sebuah *subnet*. Bekerja secara acak dan mencocokkannya dengan kumpulan *rule* serangan yang sudah disimpan pada *library*. Ketika NIDS berhasil mendeteksi serangan atau perilaku yang abnormal, peringatan akan dikirim ke administrator jaringan.

b. NNIDS (*Network Node Intrusion Detection System*), sedikit meirp dengan NIDS namun NNIDS hanya bekerja pada satu host saja tidak untuk satu jaringan. Contoh penggunaan NNIDS adalah pada VPN, yaitu dengan memeriksa *traffic* ketika sudah terdekripsi. Dengan cara ini dapat diketahui apakah seseorang sedang mencoba merusak sebuah VPN *server*.

c. HIDS (*Host Based Intrusion Detection System*), mengambil *snapshot* dari sistem yang dimiliki dan

mencocokkannya dengan *snapshot* yang sudah diambil sebelumnya. Bila sebuah *system files* penting telah termodifikasi atau terhapus, sebuah peringatan akan dikirimkan ke administrator. Contoh penggunaannya adalah pada sebuah mesin yang bersifat *mission critical*, sehingga tidak boleh ada perubahan terhadap konfigurasinya.

2.2 IDS Berbasis Anomali

Anomaly pada dasarnya adalah mencari sebuah data yang menyimpang dari sekumpulan data normal. IDS yang berbasis *anomaly* menggabungkan metode analisis dan statistik untuk mengenali penyimpangan tersebut [2]. IDS yang berbasis pada *anomaly* bersifat lebih fleksibel, karena dapat mengenali pola serangan baru tanpa harus memperbaharui basis data pola serangan. IDS yang berbais pada *anomaly* memiliki sebuah kecerdasan buatan yang mampu mendeteksi dan mengenali sebuah serangan.

Kelemahan dari metode anomali ini adalah kemungkinan terjadinya salah identifikasi pada data yang diolah, juga ada kemungkinan terjadi kesalahan pada data normal yang menyebabkan aplikasi tidak dapat mengenali serangan.

2.3 Jpcap

Jpcap [3] adalah kumpulan kelas-kelas Java yang menyediakan *interface* dan sistem untuk *packet capture* pada jaringan. Dengan bantuan Jpcap, para pengembang pada khususnya yang menggunakan bahasa pemrograman Java dapat membuat aplikasi yang memiliki kegunaan *packet capture*. Pada Gambar 2.1 adalah contoh penggunaan Jpcap pada suatu java *class*.

Gambar 2.1 Contoh penggunaan Jpcap

Salah satu kegunaan Jpcap yang sering dijumpai adalah *offline capture*. Pengertian *offline capture* adalah pembacaan *dump file* yang didapatkan dari hasil aktual suatu *data traffic*. *Offline capture* dilakukan untuk menganalisa *data traffic* yang sudah disimpan untuk tujuan tertentu antara lain, menganalisa kebiasaan pengguna, mendeteksi anomali yang mungkin terlewat dari pengawasan, dan lain sebagainya.

```
1  JpcapCaptor captor =
   JpcapCaptor.openFile("inside.tcpdump");
2  while(true){
3   Packet packet=captor.getPacket();
4   if(packet == null || packet == Packet.EOF) break;
5   System.out.println(packet);
6  }
7  captor.close();
```

Gambar 2.2 Kode sumber penggunaan Jpcap untuk *offline capture*

```
1399207224:213524
1399207224:266954 /10.151.36.24->/239.255.255.250 protocol(17) priority(0)  hop(1)
1900
1399207224:325480 /fe80:0:0:0:e10a:9351:be2f:5683->/ff02:0:0:0:0:0:1:2 protocol(17)
hop(1) UDP 546 > 547
1399207224:482300 /10.151.36.22->/10.151.36.3 protocol(17) priority(0)  hop(128)  c
1399207224:922008 /10.151.36.1->/224.0.0.10 protocol(88) priority(6)  hop(2)  offse
1399207225:181347 /10.151.36.29->/255.255.255.255 protocol(17) priority(0)  hop(128
> 17500
1399207225:183822 /10.151.36.29->/10.151.36.255 protocol(17) priority(0)  hop(128)
17500
1399207225:187066 /fe80:0:0:0:b0b8:8c4c:3dc0:d9d3->/ff02:0:0:0:0:0:1:2 protocol(17)
hop(1) UDP 546 > 547
```

Gambar 2.3 Contoh keluaran *offline capture*

2.4 N-Gram

Pada dasarnya, model N-Gram [4] adalah model probabilistik yang awalnya dirancang oleh ahli matematika dari Rusia pada awal abad ke-20 dan kemudian dikembangkan untuk memprediksi *item* berikutnya dalam urutan *item*. Item bisa berupa huruf / karakter, kata, atau yang lain sesuai dengan aplikasi. Salah satunya, model *n-gram* yang berbasis kata digunakan untuk memprediksi kata berikutnya dalam urutan kata tertentu. Dalam arti bahwa sebuah *n-gram* hanyalah sebuah wadah kumpulan kata dengan masing-masing memiliki panjang *n* kata. Sebagai contoh, sebuah *n-gram* ukuran 1 disebut sebagai *unigram;* ukuran 2 sebagai *bigram*; ukuran 3 sebagai *trigram*, dan seterusnya.

Pada pembangkitan karakter, *N-gram* terdiri dari *substring* sepanjang *n* karakter dari sejumlah *string* dalam definisi lain *n-gram* adalah potongan sejumlah *n* karakter dari sebuah *string*. Metode *n-gram* ini digunakan untuk mengambil potongan-potongan karakter huruf sejumlah *n* dari sebuah kata secara kontinuitas dibaca dari teks sumber sehingga akhir dari dokumen. Sebagai contoh : kata "TEXT" dapat diuraikan ke dalam beberapa *n-gram* berikut :

uni-gram : T, E, X, T
bi-gram : TE, EX, XT
tri-gram : TEX, EXT
quad-gram : TEXT, EXT_
dan seterusnya.

Sedangkan pada pembangkit kata, metode *n-gram* ini digunakan untuk mengambil potongan kata sejmlah *n* dari sebuah rangkaian kata (kalimat, paragraf, bacaan) yang secara kontinuitas dibaca dari teks sumber hingga akhir dari dokumen. Sebagai contoh : kalimat "saya dapat melihat cahaya itu." Dapat diuraikan ke dalam beberapa *n-gram* berikut :

uni-gram : saya, dapat , melihat, cahaya, itu
bi-gram : saya dapat, dapat melihat, melihat cahaya, cahaya itu
tri-gram : saya dapat melihat, dapat melihat cahaya, melihat cahaya itu_

dan seterusnya.

Salah satu keunggulan menggunakan *n-gram* dan bukan suatu kata utuh secara keseluruhan adlah bahwa *n-gram* tidak terlalu sensitif terhadap kesalahan penulisan yang terdapat pada suatu dokumen.

2.5 Simplified Mahalanobis Distance

Mahalanobis distance [5] adalah sebuah metode statistika untuk menghitung jarak antara titik P dan distribusi D. Prinsip Mahalanobis *Distance* adalah mengitung jarak di ruang multidimensional antara sebuah pengamatan dengan pusat dari semua pengamatan. Pada Tugas Akhir ini Mahalanobis *Distance* digunakan untuk menghitung jarak antara distribusi *byte* karakter dari payload baru terhadap model yang ada pada data *training*. Semakin jauh jaraknya, semakin besar kemungkinan payload ini tidak normal.

Mahalanobis *distance* dari sebuah payload baru dapat dihitung jika sistem sudah mempunyai data *training*. Selanjutnya menghitung rata-rata dan standar deviasi dari model yang ada pada data *training*. Untuk menghitung rata-rata dari model yang ada pada data *training* dapat dilihat pada persamaan (2.2). Sedangkan untuk menghitung standar deviasi dari model yang ada pada data *training* dapat dilihat pada persamaan (2.4). Setelah selesai

menghitung rata-rata dan standar deviasi dari model yang ada pada data *training* baru dapat menghitung jarak mahalanobis dari payload baru dengan menggunakan persamaan (2.1). Format data kasar yang ada pada Mahalanobis *disatance* dapat dilihat pada Tabel 2.1.

$$d(x,\overline{y}) = \sum_{i=0}^{n-1} (|x_i - \overline{y_i}|/(\overline{\sigma_i} + \alpha)) \qquad (2.1)$$

dimana,

d = jarak mahalanobis
x_i = variable ke-i dari *payload* baru
$\overline{y_i}$ = rata-rata variable ke-i dari model data *training*
$\overline{\sigma_i}$ = standar deviasi variable ke-i dari model data *training*
α = *smoothing factor*

Tabel 2.1 Format data kasar didalam Mahalanobis *Distance*

	Variabel (karakteristik)						
Object	X_1	X_2	…	X_i	…	X_{p-1}	X_p
1	.	.	…	.	…	.	.
2	.	.	…	.	…	.	.
3	.	.	…	.	…	.	.
.	.	.	…	.	…	.	.
.	.	.	…	.	…	.	.
.	.	.	…	.	…	.	.
K	X_{k1}	X_{k2}	…	X_{ki}	…	$X_{k,p-1}$	$X_{k,p}$
.	.	.	…	.	…	.	.
.	.	.	…	.	…	.	.
.	.	.	…	.	…	.	.
N	X_{N1}	X_{N2}	…	X_{Ni}	…	$X_{N,p-1}$	$X_{N,p}$
Average	$\overline{X_1}$	$\overline{X_2}$	…	$\overline{X_i}$	…	$\overline{X_{p-1}}$	$\overline{X_p}$
Standar deviation	S_1	X_1	…	X_1	…	X_1	X_1

Persamaan untuk mencari rata-rata, yaitu:

$$\overline{X_i} = \frac{1}{N} \sum_{k=1}^{N} X_{ki} \quad (2.2)$$

dimana,

$\overline{X_i}$ = rata-rata variabel ke-i
N = jumlah object model
X_{ki} = nilai variabel ke-i

Persamaan untuk mencari nilai standar deviasi, yaitu:

$$S_i = \sqrt{\frac{\sum_{k=1}^{N} (X_{ki} - \overline{X}_i)^2}{N-1}} \quad (2.3)$$

dimana,

S_i = standar deviasi variabel ke-i
X_{ki} = nilai dari variabel ke-i
$\overline{X_i}$ = rata-rata variabel ke-i
N = jumlah object model

2.6 *Incremental Learning*

Incremental Learning merupakan proses untuk memperbaharui nilai rata-rata dan standar deviasi dari model yang ada pada data *training* ketika menambahkan payload baru. Proses ini diperlukan untuk meningkatkan akurasi dari setiap model ketika ditambah data sampel baru.

Untuk menghitung Mahalanobis *distance* versi *Incremental Learning* diperlukan rata-rata dan standar deviasi dari masing-masing karakter ASCII untuk setiap sampel baru yang dihitung. Untuk menghitung rata-rata dari sebuah karakter dapat dilihat pada persamaan (2.3). Selanjutnya agar dapat memperbaharui nilai rata-rata dari model yang ada pada data *training*, diperlukan jumlah sampel yang telah dihitung sebelumnya [6]. Untuk menghitung nilai rata-rata yang baru dapat dilihat pada persamaan (2.4).

Sedangkan untuk mengitung standar deviasi yang baru diperlukan rata-rata dari x_i^2 pada model sebelumnya. Untuk menghitung standar deviasi yang baru dapat dilihat pada persamaan (2.5).

Persamaan untuk menghitung rata-rata baru dari model yang diamati, yaitu:

$$\overline{\boldsymbol{x}} = \frac{\overline{\boldsymbol{x}} \times \boldsymbol{N} + \boldsymbol{x}_{\boldsymbol{N+1}}}{\boldsymbol{N+1}} = \overline{\boldsymbol{x}} + \frac{\boldsymbol{x}_{\boldsymbol{N+1}} - \overline{\boldsymbol{x}}}{\boldsymbol{N+1}} \qquad (2.4)$$

dimana,

$\overline{\boldsymbol{x}}$ = rata-rata baru
$\boldsymbol{x}_{\boldsymbol{N+1}}$ = nilai dari variabel yang baru
$\boldsymbol{N}$ = jumlah sampel sebelumnya

Persamaan untuk menghitung standar deviasi baru dari model yang diamati, yaitu:

$$\boldsymbol{S}_i = \sqrt{\frac{(\boldsymbol{n+1}) \times (\sum_{i=1}^{n} \boldsymbol{x}_i{}^2 + \boldsymbol{x}_{n+1}{}^2) - (\sum_{i=1}^{n} \boldsymbol{x}_i + \boldsymbol{x}_{n+1})^2}{(\boldsymbol{n+1})\boldsymbol{n}}} \qquad (2.5)$$

dimana,

S_i = standar deviasi variabel ke-i
x_i = nilai dari variabel ke-i
x_{n+1} = nilai dari variabel yang baru
n = jumlah object model

2.7 DARPA 1999

Subbab ini akan menjelaskan data set yang nantinya akan digunakan untuk data *training* dan menguji aplikasi ini. Data set yang digunakan adalah DARPA 1999 [7]. Data set ini berisi paket-paket hasil tangkapan selama 24 jam dalam lima minggu. Penelitian akan data set ini dilakukan oleh The Cyber System and Technology Group dari MOT Lincoln Laboratory.

DARPA 1999 memiliki banyak contoh srangan. Data set ini merupakan dalah satu yang *dump file* dengan jenis serangan terlengkap sehingga data set ini sering difunakan untuk penelitian khususnya penelitian yang berhubungan dengan IDS (*Intrusion Detection System*). Beberapa publikasi ilmiah yang menggunakan data set ini antara lain:

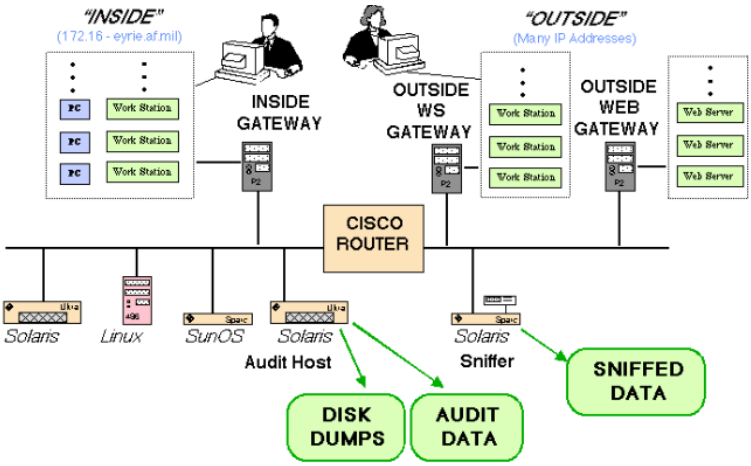
a. Richard P. Lippmann, Robert K. Cunningham, David J. Fried, Issac Graf, Kris R. Kendala, Seth E. Webster, Marc A. Zissman, "Results of the DARPA 1998 Offline Intrusion Detection Evaluation", dipresentasikan di *RAID 1999 Conference,* September 7-9, 1999, West Lafayette, Indiana.
b. Richard P. Lippmann and Robert K. Cunningham, "Using Key-String Selection and Neural Networks to Reduce False Alarms and Detect New Attacks with Sniffer-Based Intrusion DetectionSystem", dipresentasikan di *RAID 1999 Conference,* September 7-9, 1999, West Lafayette, Indiana.
c. R. K. Cunningham, R. P. Lippmann, D. J. Fried, S. L. Garfinkel, I. Graf, K. R. Kendall, S. E. Webster, D. Wyschogord, M. A. Zissman, "Evaluating Intrusion Detection System without Attacking your Friends: The DARPA 1998 Offline Intrusion Detection Evaluation", *SANS 1999*.

Contoh publikasi ilmiah diatas adalah bebrapa publikasi ilmiah yang dihasilkan oleh The Cyber Systems and Technology Group dari MIT Lincoln Laboratory, tentu masih banyak publikasi ilmiah lainnya di luar kelompok tersebut ayng menggunakan data set DARPA 1999.

2.7.1 Arsitektur Simulasi DARPA 1999

DARPA 1999 memiliki arsitektur yang cukup melambangkan aktivitas antar jaringan internal dengan eksternal. Kedua jaringan tersebut dipisahkan oleh sebuah *router* . Jaringan eksternal terdiri dari dua *workstation* yang mensimulasikan

gateway. Menuju jaringan luar secara *virtual*. Setiap *workstation* yang ada pada jaringan eksternal mensimulasikan banyak *virtual workstation* menggunakan perangkat lunak yang disediakan oleh air Force ESC. Jaringan internal meliputi *workstation* yang dijadikan korban, *workstation* ini memiliki sistem operasi bervariasi. Data didapatkan dari salah satu *workstation* yang berada di dalam dan menggunakan perangkat lunak *sniffer* untuk jaringan eksternal. Arsitektur dari DARPA 1999 terdapat pada Gambar 2.4



Gambar 2.4 Arsitektur DARPA 1999

2.7.2 Jenis – jenis Serangan dari DARPA 1999

Subbab ini akan membahas mengenai jenis-jenis serangan yang didapatkan dari uji coba ini selama lima minggu tanpa berhenti. Kategori serangan yang didapatkan terdapat pada Tabel

Jenis-jenis serangan yang didapatkan sangat banyak. Terdapat 43 jenis serangan. Serangan – serangan tersebut dapat dikategorikan menjadi 4 jenis kategori.

DoS (*Denial of Service*) adalah serangan yang bertujuan untuk mencegah server untuk melakukan pelayanan terhadap penggunanya. Pencegahan yang dimaksud adalah mengurangi

peforma hingga mematikan secara total layanan yang disediakan oleh *server*. Cara kerja DoS secara umum adalah membuat aplikasi pada *server crash,* merusak data atau membuat beban kerja dari komputer menjadi banyak. Contoh serangan DoS adalah pemanfaatan *bug* dari aplikasi, menggunakan *bad checksum*, menggunakan *spoofed address,* dan yang paling umum adalah duplikasi paket CP dengan *payload* berbeda.

Kategori selanjutnya adalah *probing*. *Probing* bertujuan untuk menemukan celah dari sistem. Cara menemukan celah tersebut adalah koneksi ke sistem secara tidak penuh missal Nmap dengan tipe *stealth scan* mengirimkan paket TCP tunggal tanpa *handshaking.*

Dua kategori terakhir adalah R2L (remote to local) dan U2R (*User to Root*). R2L adalah usaha untuk melakukan akses sebagai *rootuser* dari koneksi yang dilakukan dari jarak jauh atau udaha untuk mencari kelemahan dari sistem secara koneksi jarak jauh. U2R adalah usaha untuk mendapatkan akses sebagai *rootuser* dari dalam sistem, dalam hal ini penyerang akan masuk ke salam sistem terlebih dahulu sebagai pengguna biasa.