ANALISA PENGGUNAAN SUMBER BAGI SIMULASI SERANGAN TERAGIH KE ATAS HOS SISTEM PENGESANAN PENCEROBOHAN

ALI YUSNY BIN DAUD

Laporan projek ini dikemukakan sebagai
memenuhi sebahagian daripada syarat penganugerahan
ijazah Sarjana Sains (Sains Komputer – Keselamatan Maklumat)

Fakulti Sains Komputer Dan Sistem Maklumat
Universiti Teknologi Malaysia

FEBRUARI, 2002

UNIVERSITI TEKNOLOGI MALAYSIA

BORANG PENGESAHAN STATUS TESIS υ

JUDUL : ANALISA PENGGUNAAN SUMBER BAGI SIMULASI SERANGAN TERAGIH KE ATAS HOS SISTEM PENGESANAN PENCEROBOHAN

SESI PENGAJIAN : 2001/2002

Saya ALI YUSNY BIN DAUD
(HURUF BESAR)

Mengaku membenarkan tesis (~~PSM~~/Sarjana/~~Doktor Falsafah~~)* ini disimpan di Perpustakaan Universiti Teknologi Malaysia dengan syarat-syarat kegunaan seperti berikut :-

1. Tesis adalah hakmilik Universiti Teknologi Malaysia
2. Perpustakaan Universiti Teknologi Malaysia dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **Sila tandakan ( )

☐	SULIT	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
☐	TERHAD	(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)
☑	TIDAK TERHAD	

Disahkan oleh

(TANDATANGAN PENULIS)

(TANDATANGAN PENYELIA)

Alamat Tetap :
BATU 6 ½ JALAN KAKI BUKIT
02400 BESERI
PERLIS

PM DR MOHD AIZAINI MAAROF
Nama Penyelia

Tarikh : 4 FEBRUARI 2002

Tarikh : 4 FEBRUARI 2002

CATATAN :
* Potong yang tidak berkenaan
** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali sebab dan tempoh tesis ini perlu dikelaskan sebagai SULIT atau TERHAD
υ Tesis dimaksudkan sebagai tesis bagi Ijazah Doktor Falsafah dan Sarjana secara penyelidikan, atau disertasi bagi pengajian secara kerja kursus dan penyelidikan, atau Laporan Projek Sarjana Muda (PSM)

"Saya akui bahawa saya telah membaca karya ini dan pada pandangan saya karya ini adalah memadai dari segi skop dan kualiti untuk tujuan penganugerahan ijazah Sarjana Sains (Sains Komputer – Keselamatan Maklumat)".

Tandatangan : .....

Nama Penyelia : PM DR MOHD AIZAINI MAAROF

Tarikh : 4 FEBRUARI 2002

"Saya akui karya ini adalah hasil kerja saya sendiri kecuali nukilan dan ringkasan yang tiap-tiap satunya telah saya jelaskan sumbernya".

Tandatangan : 

Nama Penulis : ALI YUSNY BIN DAUD

Tarikh : 4 FEBRUARI 2002

DEDIKASI

Ibu tersayang……. "Terimakasih atas doa, pengorbanan dan jasa mendidikku menjadi insan berjaya". Ayah yang dihormati……."Terima kasih kerana bimbinganmu untuk haluan yang menjadikanku seperti sekarang". Keluargaku……."Doa, kasih-sayang, pengorbanan, dan sokongan tidak akan ku lupai sepanjang hayat." Teman……."Terima kasih atas segala bantuan dan pertolongan. Berharapkan jerit perih akan bertukar kejayaan yang menjadi milik kita semua. Semoga mendapat kejayaan hidup dunia dan akhirat berkat limpahan rahmat daripada Allah S.W.T. Amin.

PENGHARGAAN

Dengan nama Allah Yang Maha Pemurah dan Penyayang, lagi Maha Mengasihani. Selawat dan salam ke atas junjungan besar Nabi Muhammad s.a.w. Alhamdullilah, segala puji-pujian dan kesyukuran dipanjatkan kepada Allah S.W.T Tuhan sekalian alam kerana dengan rahmat, keizinan dan ilham yang diberikanNya, projek ini berjaya di hasilkan.

Ucapan terima kasih dan penghargaan buat penyelia projek, Profesor Madya Dr Mohd Aizaini Maarof dan pensyarah, En Mohd Yazid Idris di atas segala idea, cadangan, teguran, sokongan dan bimbingan yang diberikan.

Ucapan terima kasih juga ditujukan kepada semua para panel penilai yang sedia meneliti dan memberi komen dalam memperbaiki mutu tesis projek ini.

Penghargaan kepada pihak UUM di atas kesudian menaja dan meluluskan pengajian di peringkat Ijazah Sarjana Sains (Sains Komputer – Keselamatan Maklumat) ini.

ABSTRAK

Bidang penyelidikan dalam keselamatan maklumat merupakan satu bidang kajian yang kritikal untuk dipertahankan. Ini adalah supaya tidak ada maklumat penting jatuh kepada pihak musuh atau kepada pihak yang tidak berhak untuk mengetahuinya. Sistem Pengesanan Pencerobohan (IDS) merupakan satu peralatan yang membantu dalam menguatkan sistem pertahanan daripada ancaman ini. Ancaman atau serangan yang dibuat secara teragih dan dalam pelbagai bentuk menyukarkan IDS untuk membuat pengesanan serta melibatkan penggunaan sumber yang banyak. Kekurangan sumber pada hos IDS akan menyebabkan ia gagal untuk melakukan pengesanan pencerobohan dan seterusnya mengatasi serangan tersebut. Tujuan utama kajian ini adalah untuk menganalisa serangan secara teragih terhadap sebuah hos IDS dan melihat bagaimana penggunaan sumber dalam mengesan serangan tersebut. Teknik simulasi telah digunakan untuk mewakili keadaan dunia sebenar dalam pengesanan pencerobohan. Simulasi telah digunakan kerana di dalam dunia sebenar, penganalisaan serangan melibatkan skala yang besar dan kos yang amat tinggi. Kajian ini akan memfokuskan kepada analisa simulasi serangan dan menggunakannya untuk membantu pembangunan IDS dalam membuat pengesanan yang lebih berkesan.

ABSTRACT

Research field in information security is a critical research to defend of. It's trivial for protecting important information from the enemy or to unauthorized person. Intrusion Detection System (IDS) is a tool in supporting to defend from the threats. Distributed threats or attacks in various form makes detection by IDS very complicated and involving a lot of resource usage. Lack of resources in the IDS will affect intrusion detection and fail to overcome the attack. The purpose of this research is to analyse distributed attacks to a single IDS host and resource usage when it detecting the attacks. Simulation technique is used to present reality in intrusion detection. Simulation is applicable because in reality, intrusion's analysis involved large scale and high costing. The research will focus to the analysis of the simulation and using the result for IDS to make an efficient detection.

KANDUNGAN

SENARAI JADUAL

SENARAI RAJAH

SENARAI SIMBOL

BPF	-	Penapis paket Berkeley (*Berkeley Packet Filter)*
CERT	-	Pasukan Tindakbalas Kecemasan Komputer (*Computer Emergency Response Team*)
CPU	-	Unit pemprosesan pusat (*central processing unit*)
DARPA	-	Agensi Projek Penyelidikan Pertahanan Lanjutan
DoS	-	Program penafian perkhidmatan (*Denial of Services*)
FAT	-	Jadual Peruntukan Fail (*File Allocation Table*)
GB	-	Gigabait
IDS	-	Sistem Pengesanan Pencerobohan
IRC	-	*Internet Relay Chat*- program *chatting* yang popular dan menjadi port yang terdedah untuk serangan.
IRT	-	Pasukan tindakbalas kejadian (*incident response team*)

kb/s	-	kilobait per saat
TCP	-	Protokol penghantaran komunikasi (*Transfer communication protocol*)
ICMP	-	Protokol kawalan mesej Internet (*Internet Control Message Protocol*)
IP	-	Protokol Internet (*Internet protocol*)
MB	-	Megabait
MHz	-	Megahertz
RAM	-	Memori capaian rawak

SENARAI LAMPIRAN

BAB 1

PENGENALAN

Dunia telah dilanda oleh perkembangan teknologi maklumat yang mendadak semenjak Internet diperkenalkan. Bertitik tolak daripada itu, pelbagai maklumat telah dimuatkan ke ruang maya untuk kegunaan para pengguna. Namun malangnya, ancaman terhadap pengguna maklumat turut berkembang dengan pesat apabila program-program yang mengakibatkan kerosakan dan keburukan juga dimasukkan ke dalamnya.

Masyarakat dunia kini juga mula menggunakan Internet sebagai alat transaksi perniagaan iaitu e-dagang. Pertumbuhan yang mendadak ini menyebabkan sistem komputer dan perhubungan antara rangkaian, menjadi satu teknologi yang amat penting. Kebergantungan organisasi dan juga individu terhadap maklumat yang tersimpan serta komunikasi yang dilakukan menerusi rangkaian meningkat dengan begitu drastik sekali.

Selari dengan perkembangan itu, rangkaian komputer telah menjadi mangsa kepada serangan atau ancaman daripada penggodam (Mantha *et al.*, 1999). Keselamatan maklumat terus menjadi isu yang hangat untuk diperkatakan dan dipertahankan. Sistem komputer dan rangkaiannya perlu diperkuatkan untuk menahan daripada pelbagai ancaman yang sentiasa mencari-cari peluang untuk menceroboh masuk. Di sinilah fungsi keselamatan maklumat bagi menangani masalah sebegini. Menurut Stallings dalam bukunya *Cryptography And Network Security* (1995), keselamatan maklumat boleh didefinisikan sebagai sekumpulan

peralatan yang direkabentuk untuk mempertahankan data dan menghalang penggodam daripada menceroboh sistem komputer.

Pfleeger (1997) menyatakan dalam bukunya *Security In Computing* bahawa keselamatan maklumat mempunyai tiga matlamat untuk dicapai.

1. Kerahsiaan (*confidentiality*). Aset di dalam sistem komputer hanya boleh dicapai oleh pihak yang berhak sahaja. Jenis capaian adalah capaian jenis baca (*read*) iaitu bacaan, paparan, cetakan atau sekadar mengetahui kewujudan objek tersebut.

2. Keutuhan (*integrity*). Aset hanya boleh dibuat perubahan oleh pihak yang sah dengan cara yang juga sah. Dalam konteks ini perubahan yang dibuat termasuklah penulisan, pengubahan dokumen, pengubahan status, memadam dan membina maklumat yang baru.

3. Kedapatan (*availability*). Aset hendaklah boleh dicapai oleh pihak yang sah. Pihak yang sah ini tidak boleh dihalang daripada mencapai objek atau maklumat yang mana dia mempunyai hak ke atasnya.

Daripada kacamata seorang atau sekumpulan penggodam komputer, mereka sentiasa mencari titik kelemahan di dalam sistem yang membolehkan mereka memasuki sistem dan melakukan kerosakan. Ini adalah mengikut prinsip yang dinyatakan oleh Pfleeger (1997) di dalam bukunya *Security In Computing* dimana "Prinsip Pencerobohan Termudah" menyatakan bahawa seseorang penceroboh mestilah diandaikan menggunakan segala jenis cara pencerobohan. Ia tidak semestinya cara yang paling selalu digunakan ataupun cuba mengatasi sistem keselamatan yang paling kuat.

Salah satu cara yang paling sesuai untuk menghadapi masalah ini adalah dengan cuba melihat masalah ini dari sudut penggodam itu sendiri. Kajian ini dibuat bertujuan untuk mencari bagaimana caranya untuk mengatasi sistem keselamatan yang digunakan. Oleh itu simulasi dibuat untuk mengandaikan serangan yang dibuat oleh penggodam terhadap sistem komputer. Analisa yang dihasilkan digunakan pula

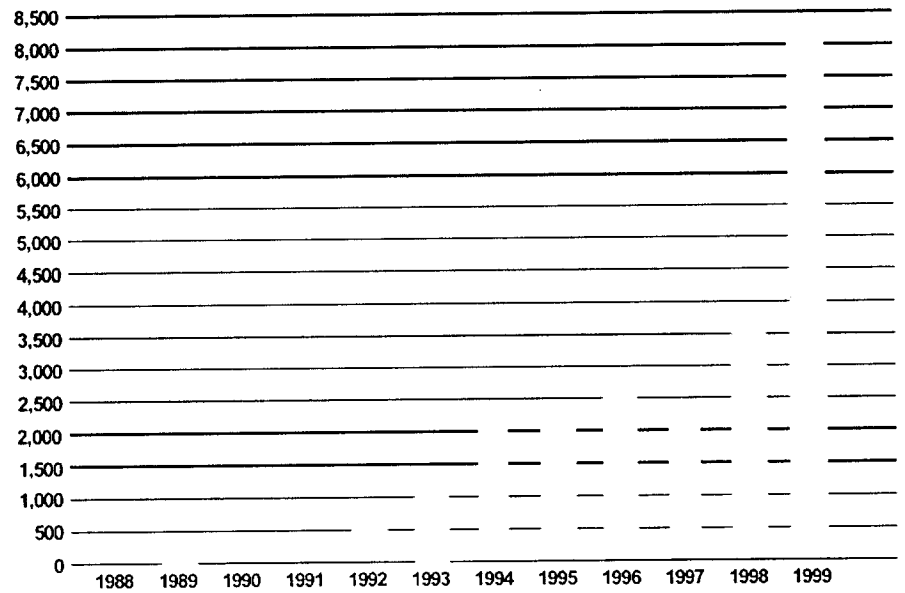
untuk meramal ciri serangan dan menghasilkan pengesanan yang lebih baik serta berkesan.

1.1 Latarbelakang Masalah

Menurut Allen *et al.* (2000), pada sekitar tahun 1980an, penceroboh adalah daripada kalangan pakar sistem. Mereka mempunyai kepakaran yang tinggi dan berupaya mencipta kaedah tersendiri untuk menceroboh sistem komputer yang lain. Penggunaan alatan otomatik dan skrip yang diubahsuai telah digunakan. Namun pada hari ini, sesiapa sahaja boleh menceroboh sesebuah rangkaian disebabkan oleh tersebarnya serta kemudahan untuk mendapatkan alatan pencerobohan dan pengubahsuaian skrip yang mempunyai kaedah serangan di dalam Internet.

Sementara itu penceroboh yang berpengalaman menjadi semakin bijak disebabkan oleh bantuan kecanggihan jenis-jenis serangan yang ada. Penggodam yang baru untuk menceburkan diri dalam pencerobohan sistem komputer tidak memerlukan pengetahuan yang tinggi untuk meniru atau melancarkan kaedah serangan yang sama ke atas sistem komputer. Mereka hanya perlu menggunakan perisian serangan yang senang didapati dan senang digunakan di dalam Internet. Rajah 1.1 menunjukan peningkatan insiden pencerobohan sepertimana yang dilaporkan oleh *Computer Emergency Response Team* atau CERT bagi tahun 1988 hinggalah 1999 (Carver *et al.*, 2000).

Ada banyak sebab yang menjadikan peningkatan serangan menjadi sebegini. Ini termasuklah peningkatan perhubungan dan peningkatan kesukaran; pertambahan kedapatan untuk maklumat dan skrip serangan melalui Internet serta kebergantungan kepada perkhidmatan rangkaian teragih. Kesukaran ini meningkatkan keperluan untuk mengkaji serangan dan pengesanan serangan teragih dan selari (Mantha *et al.*, 1999).



Rajah 1.1: Peningkatan dalam insiden mengikut CERT(Carver *et al.*, 2000)

Serangan dan pencerobohan boleh dilihat daripada pelbagai perspektif. Cara yang paling banyak digunakan adalah daripada perspektif penceroboh dan juga perspektif mangsa pencerobohan. Bagi setiap perspektif, kriteria yang berlainan digunakan untuk menyatakan serangan itu berjaya atau tidak. Secara umumnya pencerobohan telah berlaku apabila serangan adalah berjaya daripada perspektif mangsa atau secara mudahnya mangsa telah mengalami kehilangan aset atau mendapat akibat daripadanya. Serangan yang berjaya adalah disebabkan adanya kelemahan dalam sistem mangsa yang dieksploit oleh penceroboh dengan objektif yang tersendiri (Allen *et al.*, 2000).

Proses pencerobohan bermula apabila penceroboh mengambil langkah-langkah perlu untuk memenuhi objektifnya. Komponen asas pencerobohan mengambil kesempatan daripada satu atau lebih kelemahan dalam sistem mangsa dengan menggunakan peralatan ataupun skrip eksploitasi.

Kelemahan yang dieksploit daripada proses ini boleh disebabkan oleh adanya masalah dalam sesuatu perisian sehinggalah kepada kelemahan di dalam struktur organisasi yang membolehkan serangan mendapatkan maklumat sensitif atau kata laluan pihak mangsa. Proses pencerobohan berakhir apabila sebahagian atau semua objektif penceroboh dipenuhi atau apabila penceroboh menamatkan serangan.

Membina dan meningkatkan teknik pengesanan pencerobohan komputer adalah bidang penyelidikan yang berterusan dilakukan oleh Agensi Projek Penyelidikan Pertahanan Lanjutan (DARPA), di mana ia telah membiayai lebih 20 projek yang berlainan dalam organisasi atau universiti (Durst *et al.*, 1999).

Langkah pertama dalam mempertahankan serangan ke atas sistem komputer adalah dengan mengetahui samada serangan telah berlaku atau bila ia bermula, atau dengan kata lain mengesan pencerobohan tersebut. Pengesanan pencerobohan adalah bidang yang cuba mengesan pencerobohan (samada dari penggodam luaran atau dalaman), terutamanya ketika perlaksanaan serangan (Stillerman *et al.*, 2000).

Satu masalah yang dihadapi oleh pengurus rangkaian komputer ialah apabila terlalu banyak ancaman yang tidak diketahui dan juga ancaman yang tidak berakhir. Memburukkan lagi keadaan adalah apabila perisian untuk menceroboh ke dalam rangkaian komputer boleh didapati dengan mudah daripada Internet malahan percuma dan lengkap dengan modul arahan. Penggodam sentiasa mencipta pelbagai serangan baru secara berterusan dan menyebarkannya ke dalam Internet.

Masalah ini begitu berleluasa, tidak sahaja dialami oleh kerajaan malah banyak organisasi cuba menangani perisikan industri (usaha mencuri maklumat daripada pesaing) setiap hari untuk menjaga maklumat dagangan yang sulit. Penggodam muda yang dilihat sebagai tidak merbahaya sebagaimana mereka yang telah berpengalaman, masih boleh merosakkan sistem dan juga pertahanan komputer mangsa. Tambahan lagi ada juga pihak dalaman organisasi sendiri yang mahu menceroboh daripada dalam organisasi.

Teknologi seperti ActiveX, Java dan juga teknologi lain yang seumpamanya (*commercial off-the-shelf technology*) membantu penceroboh melakukan aktiviti seolah-olah seorang pengguna yang sah. Pelbagai lagi ancaman yang muncul di mana pada setiap masa dan setiap ketika, ada sahaja ancaman baru yang terbina apabila hadirnya penceroboh baru yang lebih bijak. Sudah tiba masanya untuk melihat masalah pencerobohan ini dengan mengandaikan kita seperti penceroboh itu sendiri (Allen *et al.*, 2000).

Pengesanan pencerobohan diperlukan kerana "dinding api" *(firewall)* tidak dapat menyediakan perlindungan keseluruhan kepada pencerobohan. Pengalaman telah mengajar kita supaya tidak hanya bergantung kepada hanya satu tembok pertahanan atau teknik untuk berlindung daripada serangan.

"Dinding api" *(firewall)* bertindak sebagai penapis maklumat yang baik dan menahan banyak serangan sebelum memasuki rangkaian organisasi. Walau bagaimanapun ia mempunyai kelemahan apabila adanya ralat dalam konfigurasi atau polisi keselamatan yang tidak didefinisikan dengan betul dalam sesebuah organisasi. Ia tidak berupaya untuk melindungi daripada kod penghantaran jahat (*malicious code*), serangan dalaman, dan ketidakselamatan modem.

Kajian pengesanan pencerobohan telah dibuat sejak 20 tahun dahulu, tetapi ia masih lagi berada di tahap yang rendah. IDS yang sedia ada tidak berfungsi dengan baik untuk mengesan sumber dan penggodam yang boleh menceroboh melalui pelbagai cara yang terkini. Selain daripada itu, IDS tidak dapat untuk mengecam pencerobohan dan seringkali memberikan amaran palsu (*false alarm*).

Kelemahan IDS yang dihasilkan kini dilihat disebabkan oleh kecenderungan untuk menumpukan kepada pencerobohan secara individu sahaja (Goan, 1999). Oleh itu serangan yang dibuat secara teragih menjadi lebih sukar untuk dikesan. Sebelum ini juga, kajian teragih yang dibuat kebanyakannya adalah berkaitan dengan serangan penafian perkhidmatan teragih tanpa melihat serangan-serangan yang lain (Kargl, 2001). Memburukkan lagi keadaan, senario serangan begini mengakibatkan penggunaan sumber ke tahap kritikal yang mengancam prestasi IDS itu sendiri (Puketza *et al.*, 1994).Oleh itu dalam kajian ini serangan teragih daripada pelbagai jenis dan secara selari dilakukan untuk penganalisaan penggunaan sumber.

1.2 Pernyataan Masalah

Pernyataan masalah bagi kajian ini boleh disimpulkan sebagai satu persoalan utama iaitu:

Apakah kesan serangan teragih secara selari terhadap sebuah hos sistem pengesanan pencerobohan (IDS) dalam mengesan pencerobohan terutamanya dalam penggunaan sumber IDS?

Kajian ini memfokuskan kepada analisa serangan teragih secara selari yang dibuat secara simulasi ke atas sebuah hos IDS. Ia menekankan tentang keberkesanan IDS dalam mengesan serangan yang dibuat secara teragih dengan jenis serangan yang dipelbagaikan. Ia seterusnya melihat jumlah sumber yang diperlukan dalam membuat pengesanan serangan ini.

1.3 Matlamat Kajian

Matlamat kajian ini ialah membuat analisa serangan teragih secara selari ke atas sebuah hos sistem pengesanan pencerobohan bagi mendapatkan maklumat dalam penggunaan sumber oleh IDS, ciri-ciri serangan, kelemahan pengesanan dan cadangan untuk meningkatkan prestasinya.

1.4 Objektif Kajian

Objektif utama bagi kajian ini ialah untuk:

1. Membuat pemerhatian dalam persekitaran simulasi sebagai andaian serangan ke atas komputer dalam situasi sebenar.
2. Melihat penggunaan sumber yang diperlukan oleh IDS untuk membuat pengesanan daripada serangan individu, teragih dan pelbagai serangan.
3. Melihat perbezaan serangan yang dibuat secara teragih dengan serangan individu.
4. Membuat perbandingan antara beberapa serangan yang sama dan juga serangan yang berlainan dalam persekitaran teragih.

1.5 Skop Kajian

Kajian ini mengkaji beberapa serangan teragih secara selari terhadap sebuah komputer yang mempunyai sistem pengesanan pencerobohan. Serangan ini dianalisa daripada segi penggunaan sumber untuk mengesan dan hasilnya dapat digunakan sebagai cadangan dalam meningkatkan prestasi sistem pengesanan pencerobohan.

Bagi proses pengesanan, kajian ini hanya menggunakan sistem pengesanan yang berasaskan corak atau tandatangan. IDS yang digunakan tidak menggunakan pengesanan secara anomali. Selain itu kajian ini hanya menggunakan sistem pengesanan hos yang berada dalam platform Windows NT sahaja.

Andaian telah dibuat bahawa serangan yang dibuat adalah mewakili serangan teragih dalam situasi sebenar. Serangan-serangan juga adalah terhad kepada jenis serangan yang terpilih sahaja dan tidak melibatkan semua jenis serangan.

1.6 Justifikasi dan Kepentingan Kajian

Justifikasi dan kepentingan kajian ini ialah:

1. Output daripada analisa ini dapat digunakan untuk memperbaiki dan memperincikan lagi sistem pengesanan pencerobohan untuk menghadapi serangan teragih yang pelbagai.
2. Mencadangkan spesifikasi sumber yang diperlukan oleh sesebuah hos IDS untuk menghadapi serangan dalam dunia sebenar.
3. Menyediakan persekitaran komputer yang selamat daripada ancaman dan pencerobohan sama ada daripada luar ataupun dalam.
4. Simulasi yang dilakukan dapat menggambarkan keadaan sebenar dengan kos yang subjektifnya lebih murah dengan alternatif serangan teragih yang pelbagai.
5. Menekankan penggunaan set peraturan yang lebih berkesan dalam membuat pengesanan serangan.

LAMPIRAN C

Fail snort.conf

```
# $Id: snort.conf,v 1.14 2001/01/05 19:27:33 roesch Exp $
##################################################################
# This file contains a sample snort configuration. You can take the
# following steps to create your own custom configuration:
#
#  1) Set the HOME_NET variable for your network
#  2) Configure preprocessors
#  3) Configure output plugins
#  4) Customize your rule set
#
##################################################################
# Step #1: Set the HOME_NET variable:
#
#    You must change the HOME_NET variable to reflect your local
#    network. The variable is currently setup for an RFC 1918
address
#    space.
#
#    You can specify it explicitly as: var HOME_NET 10.1.1.0/24
#    or use global variable $<intname>_ADDRESS which will be always
#    initialized to IP address and netmask of the network interface
#    which you run snort at.
#
#    You can specify lists of IP addresses by separating the IPs
with commas
#    like this:
#
#    [10.1.1.0/24,192.168.1.0/24]
#
#    MAKE SURE YOU DON'T PLACE ANY SPACES IN YOUR LIST!
#
#
#var HOME_NET $eth0_ADDRESS

var HOME_NET 161.139.69.4/32

# Set up the external network addresses as well.  A good start may
be
# "any"...

var EXTERNAL_NET any

# Define the addresses of DNS servers and other hosts if you want to
ignore
# portscan false alarms from them...

#var DNS_SERVERS [192.168.1.1/32,10.1.1.1/32]

##################################################################
# Step #2: Configure preprocessors
#
# General configuration for preprocessors is of the form
#
```

```
#   preprocessor <name_of_processor>: <configuration_options>

# minfrag: detect small fragments
# ------------------------------
# minfrag takes the minimum fragment size (in bytes) threshold as
its
# argument. Fragmented packets at of below this size will cause an
# alert to be generated.  The functionality of this preprocessor is
# largely superceded by the defrag plugin below.

#preprocessor minfrag: 128

# defrag: defragmentation support
# ------------------------------
# IP defragmentation support from Dragos Ruiu. There are no
# configuration options at this time.

preprocessor defrag

# stream: TCP stream reassembly
# ----------------------------
# TCP stream reassembly preprocessor from Chris Cramer.  This
# preprocessor should always go after the defrag preprocessor, but
# before http_decode. The example below monitors ports 23 and 80,
has
# a timeout after 10 seconds, and will send reassembled packets of
max
# payload 16384 bytes through the detection engine. See
# README.tcpstream for more information and configuration
# options. Uncomment the following line and configure appropriately
to
# enable this preprocessor.
#
# NOTE: This code should still be considered BETA!  It seems to be
stable, but
# there are still some issues that remain to be resolved, so make
sure
# you keep an eye on your Snort sensor if you enable this plugin

# preprocessor stream: timeout 10, ports 21 23 80, maxbytes 16384

# http_decode: normalize HTTP requests
# ------------------------------------
# http_decode normalizes HTTP requests from remote machines by
# converting any %XX character substitutions to their ASCII
# equivalent. This is very useful for doing things like defeating
# hostile attackers trying to stealth themselves from IDSs by mixing
# these substitutions in with the request. Specify the port
# numbers you want it to analyze as arguments.

preprocessor http_decode: 80 8080

# portscan: detect a variety of portscans
# ---------------------------------------
# portscan preprocessor by Patrick Mullen <p_mullen@linuxrc.net>
# This preprocessor detects UDP packets or TCP SYN packets going to
# four different ports in less than three seconds. "Stealth" TCP
# packets are always detected, regardless of these settings.

preprocessor portscan: $HOME_NET 4 3 portscan.log

# Use portscan-ignorehosts to ignore TCP SYN and UDP "scans" from
```

```
# specific networks or hosts to reduce false alerts. It is typical
# to see many false alerts from DNS servers so you may want to
# add your DNS servers here. You can all multiple hosts/networks
# in a whitespace-delimited list.
#
#preprocessor portscan-ignorehosts: $DNS_SERVERS

# Spade: the Statistical Packet Anomaly Detection Engine
#-------------------------------------------------------
#
# READ the README.Spade file before using this plugin!
#
# See http://www.silicondefense.com/spice/ for more info
#
# Spade is a Snort plugin to report unusual, possibly suspicious,
packets.
# Spade will review the packets received by Snort, find those of
interest (TCP
# SYNs into your homenets, if any), and report those packets that it
believes
# are anomalous along with an anomaly score.  To enable
spp_anomsensor, you
# must have a line of this form in your snort configuration file:
#
# preprocessor spade: <anom-report-thresh> <state-file> <log-file>
<prob-mode>
#                     <checkpoint-freq>
#
# DO NOT ENABLE THIS PLUGIN UNLESS YOU HAVE READ THE README.Spade
FILE THAT
# COMES IN THIS DISTRIBUTION AND ARE COGENT OF THE PERFORMANCE
IMPACT THAT THIS
# MODULE MAY HAVE UPON YOUR NORMAL SNORT CONFIGURATION!
#
# set this to a directory Spade can read and write to store its
files
#
# var SPADEDIR .
#
# preprocessor spade: -1 $SPADEDIR/spade.rcv $SPADEDIR/log.txt 3
50000
#
# put a list of the networks you are interested in Spade observing
packets
# going to here
#
# preprocessor spade-homenet: 0.0.0.0/0
#
# this causes Spade to adjust the reporting threshold automatically
# the first argument is the target rate of alerts for normal
circumstances
# (0.01 = 1% or you can give it an hourly rate) after the first hour
(or
# however long the period is set to in the second argument), the
reporting
# threshold given above is ignored you can comment this out to have
the
# threshold be static, or try one of the other adapt methods below
#
# preprocessor spade-adapt3: 0.01 60 168
#
# other possible Spade config lines:
```

```
# adapt method #1
#preprocessor spade-adapt: 20 2 0.5
# adapt method #2
#preprocessor spade-adapt2: 0.01 15 4 24 7
# offline threshold learning
#preprocessor spade-threshlearn: 200 24
# periodically report on the anom scores and count of packets seen
#preprocessor spade-survey:  $SPADEDIR/survey.txt 60
# print out known stats about packet feature
#preprocessor spade-stats: entropy uncondprob condprob


##################################################################
# Step #3: Configure output plugins
#
# Uncomment and configure the output plugins you decide to use.
# General configuration for output plugins is of the form:
#
# output <name_of_plugin>: <configuration_options>
#
# Note that you can optionally define new rule types and associate
one
# or more output plugins specifically to that type.
#
# This example will create a type that will log to just tcpdump.
# ruletype suspicious
# {
#   type log
#   output log_tcpdump: suspicious.log
# }
#
# This example will create a rule type that will log to syslog
# and a mysql database.
# ruletype redalert
# {
#   type alert
#   output alert_syslog: LOG_AUTH LOG_ALERT
#   output database: log, mysql, user=snort dbname=snort
host=localhost
# }

# alert_syslog: log alerts to syslog
# ----------------------------------
# Use one or more syslog facilities as arguments
#
# output alert_syslog: LOG_AUTH LOG_ALERT

# log_tcpdump: log packets in binary tcpdump format
# -------------------------------------------------
# The only argument is the output file name.
#
# output log_tcpdump: snort.log

# database: log to a variety of databases
# ---------------------------------------
# See the README.database file for more information about
configuring
# and using this plugin.
#
# output database: log, mysql, user=snort dbname=snort
host=localhost
# output database: log, postgresql, user=snort dbname=snort
```

```
# output database: log, unixodbc, user=snort dbname=snort

# xml: xml logging
# ----------------
# See the README.xml file for more information about configuring
# and using this plugin.
#
# output xml: log, file=/var/log/snortxml


##################################################################
# Step #4: Customize your rule set
#
# Up to date snort rules are available at the following web sites:
#   http://www.snort.org
#   http://www.whitehats.com
#
# The snort web site has documentation about how to write your own
# custom snort rules.
#
# The rules included with this distribution generate alerts based on
# on suspicious activity. Depending on your network environment,
your
# security policies, and what you consider to be suspicious, some of
# these rules may either generate false positives ore may be
detecting
# activity you consider to be acceptable; therefore, you are
# encouraged to comment out rules that are not applicable in your
# environment.
#
# Note that using all of the rules at the same time may lead to
# serious packet loss on slower machines. YMMV, use with caution,
# standard disclaimers apply. :)
#
# The following individuals contributed many of rules in this
# distribution.
#
# Credits:
#   Ron Gula <rgula@securitywizards.com> of Network Security Wizards
#   Martin Markgraf <martin@mail.du.gtn.com>
#   CyberPsychotic <fygrave@tigerteam.net>
#   Nick Rogness <nick@rapidnet.com>
#   Jim Forster <jforster@rapidnet.com>
#   Scott McIntyre <scott@whoi.edu>
#   Tom Vandepoel <Tom.Vandepoel@ubizen.com>

include webcgi-lib
include webcf-lib
include webiis-lib
include webfp-lib
include webmisc-lib
include overflow-lib
include finger-lib
include ftp-lib
include smtp-lib
include telnet-lib
include misc-lib
include netbios-lib
include scan-lib
include ddos-lib
include backdoor-lib
include ping-lib
```