mengimbas port dalam masa 60 saat. K8 menghantar polisi yang menghalang K6 dari mengimbas port kepada semua K1-K10. Akibatnya K6 gagal untuk mengimbas port pada K2 dan tidak boleh memulakan serangan. K6 cuba untuk mengimbas K4 tetapi gagal kerana polisi masih menghalang dari K6 berbuat demikian. Semua K1-K10 telah dikemaskini dengan polisi baru yang menghalang K6 dari membuat serangan.

4.1 PENDEKATAN DINAMIK SISTEM *REDALERT*

Pendekatan dinamik Sistem *RedAlert* bermakna sistem berkebolehan dalam menghasilkan tindakan dinamik. Tindakan dinamik merupakan tindakan yang mengalami perubahan dan perubahan ini adalah ketara dan mengikut arahan daripada sistem sokongan lain seperti Snort.

Pendekatan dinamik ini penting bagi memastikan polisi dikemaskini dan keselamatan rangkaian selamat dan stabil. Pendekatan dinamik mampu memberi sumbangan yang besar dalam isu keselamatan. Jika dilihat pendekatan statik dalam kebanyakan sistem keselamatan dalam rangkaian dan juga hos, ianya tidak memberi jaminan keselamatan yang baik. Pendekatan statik bermakna polisi dalam isu keselamatan adalah statik dan tidak berubah.

Tambahan pula pendekatan secara dinamik sebenarnya bertindak secara automatik dan tindakan sebegini memudahkan pentadbir untuk memberi perhatian kepada perkara lain dalam rangkaian. Walaupun isu keselamatan merupakan isu yang penting dalam rangkaian, keselamatan rangkaian yang sering terdedah kepada penceroboh akan

menyusahkan pentadbir. Pendekatan dinamik dan juga automatik bukan sahaja dapat menghalang penceroboh dengan kadar yang cepat malahan memberi peluang kepada pentadbir rangkaian untuk membuat tugas lain.

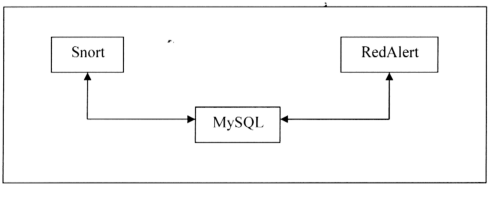
Sejak dahulu pentadbir rangkaian memeriksa rangkaian dengan menyemak skrip atau log yang disimpan oleh sistem IDS dan membuat tindakan selepas pemerhatian dan analisa dilakukan. Proses kerja ini lambat dan memberi peluang kepada penceroboh untuk menceroboh sistem. Dengan adanya sistem dinamik dan automatik, penceroboh dapat dihalang dari terus menceroboh. Walaupun tidak sepenuhnya tetapi sekurang-kurangnya dapat melindungi pelayan lain dalam rangkaian yang sama. Pengemaskinian polisi secara dinamik dan automatik berlandaskan sistem *RedAlert* memberi pendekatan baru dalam menangani isu keselamatan yang sering berubah dan sukar.

Pendekatan dinamik bergantung kepada sistem RedAlert untuk mentafsir serangan yang memerlukan tindakan segera serta memastikan setiap tindakan itu perlu atau tidak. Ini bermakna ada serangan yang merupakan serangan biasa dan tidak memerlukan tindakan lanjutan daripada sistem. Pentadbir akan mentafsir bentuk serangan ini.

4.2 SNORT DAN SISTEM *REDALERT*

Snort merupakan aplikasi yang diintegrasi bersama dengan sistem *RedAlert* untuk membolehkan sistem ini beroperasi secara baik. Penggunaan Snort adalah bertujuan untuk mendapatkan data berkaitan pencerobohan dan digunakan oleh sistem *RedAlert* untuk membuat pertimbangan.

Data pencerobohan yang diperolehi oleh Snort akan dinilai semula oleh *RedAlert*. Data kemudian dibandingkan dengan pangkalan data *RedAlert*, jika didapati ianya merupakan serangan yang perlu dihalang, *RedAlert* akan menjalankan proses seterusnya. Walaupun Snort merupakan aplikasi luaran, namun proses integrasi dengan sistem *RedAlert* membentuk satu proses yang boleh menghalang pencerobohan secara dinamik dan automatik. Snort penting bagi memastikan aplikasi *RedAlert* beroperasi dengan baik dan stabil. Secara pembentukan proses Snort digunakan sebagai peranti tambahan bagi sistem RedAlert. Ini disebabkan data yang diperolehi oleh Snort disimpan dalam pangkalan data MySQL yang turut sama digunakan oleh RedAlert.



Rajah 4.2 : Saling bertindak Snort dan RedAlert menggunakan MySQL

Saling tindakan antara RedAlert dan juga Snort adalah merujuk kepada beberapa perkara. Data Snort yang diguna oleh RedAlert adalah:

i. Log Pencerobohan

Log pencerobohan yang diperolehi daripada Snort disimpan oleh MySQL. Data ini digunakan oleh RedAlert untuk dinilai dan dianalisis untuk tindakan selanjutnya.

ii. Maklumat Paket Pencerobohan.

Maklumat paket pencerobohan bermakna maklumat mengenai paket yang mempunyai tandatangan teknik pencerobohan akan diperolehi oleh Snort dan di simpan dalam MySQL. Data ini digunakan oleh RedAlert untuk membuat polisi keselamatan baru berdasarkan kepada elemen seperti alamat sumber dan destinasi serta port yang digunakan.

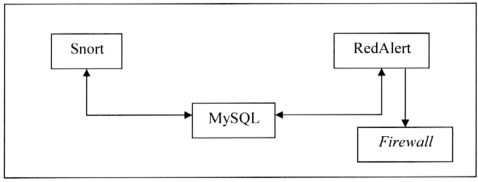
4.3 POLISI *FIREWALL* SISTEM *REDALERT*

Polisi *firewall* sistem RedAlert merujuk kepada micro-*firewall* yang digunakan oleh sistem itu sendiri. Secara khusus micro-*firewall* yang digunakan adalah iptables yang merupakan *firewall* bagi Linux Kernel 2.4x dan ke atas.

Secara umumnya penggunaan *firewall* iptables memberi lebih keselamatan kepada pengguna persendirian. Perubahan polisi *firewall* pada pengguna peribadi adalah disebabkan oleh sistem RedAlert yang bertindak secara automatik apabila Snort mengesan cubaan menceroboh sedang dilakukan oleh penceroboh. Perkaitan antara

firewall ini dengan sistem RedAlert adalah saling kaitan, sama seperti Snort dan RedAlert. Ketiga-tiga elemen ini memainkan peranan penting antara satu sama lain yang mana ketiga-tiga sistem berjalan secara serentak dan hasilnya satu tindakan dinamik dan automatik dilakukan.

Polisi *firewall* sebenarnya dibentuk oleh RedAlert dan data mengenai alamat dan destinasi sumber serta port diperolehi daripada pangkalan data Snort yang disimpan dalam MySQL. RedAlert membuat polisi baru dan polisi ini digunakan oleh Linux *firewall*. Sistem-sistem ini dijalankan secara keseluruhannya menggunakan sistem operasi Linux.



Rajah 4.3 : Tindakan Lanjutan pada *Firewall* selepas RedAlert

Tindakan RedAlert ini, yang membuat polisi baru kemudian dihantar melalui transmisi multicast kepada terminal yang mempunyai agen multicast. Kemudian proses pengemaskinian polisi baru akan berjalan. Polisi ini dihantar dalam bentuk paket multicast bersama dengan kaedah penyulitan. Agen yang menerima paket ini mempunyai arahan polisi baru bagi *firewall*. Polisi dicipta berdasarkan pangkalan data RedAlert.

Tindakan yang dilakukan ke atas *firewall* merupakan tindakan sementara dan dinamik, maka wujud dua keadaan penting iaitu keadaan penambahan dan keadaan pembatalan polisi yang wujud.

Elemen yang penting dalam menjalankan proses polisi *firewall* adalah :

i. Alamat sumber dan destinasi.

   Merujuk kepada alamat sumber penceroboh dan destinasi pencerobohan. Alamat ini digunakan untuk membina polisi yang baru.

ii. Port.

   Merujuk kepada port sumber dan port destinasi pencerobohan. Port ini digunakan untuk membina polisi yang baru.

iii. Rantaian.

   Merujuk kepada rantaian *firewall* sama ada menggunakan rantaian INPUT, OUTPUT atau FORWARD dalam menghasilkan polisi. Hanya wujud satu rantaian pada satu-satu polisi.

iv. Mod –A atau –D

   Merujuk kepada mod dalam membina polisi *firewall*. –A merujuk kepada tambahan kepada polisi sedia ada dan –D merujuk kepada membuang polisi jika berkenaan.

v. Masa

Merujuk kepada masa yang perlu diambil untuk mengekalkan polisi *firewall*, bermakna sistem menentukan masa polisi *firewall* sama ada dalam tempoh masa tertentu atau selama-lamanya.

vi. Kebenaran.

Merujuk kepada kebenaran satu polisi sama ada membenarkan laluan (ALLOW) atau tidak (DENY). Hanya digunakan sekali pada satu-satu masa.

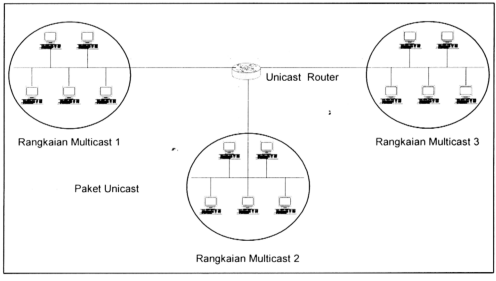
Rangka kerja proses pengemaskinian ini dilakukan cuma menggunakan proses dan pendekatan "tambah, semak dan buang". Proses pendekatan ini dilakukan dengan menetapkan polisi berkaitan dengan masa.

Apabila satu polisi dinyatakan dalam sistem dan ianya merupakan polisi baru maka ianya akan ditambah. Setelah polisi berada dalam sistem dan tempoh masa tamat, maka polisi akan disingkirkan daripada sistem. Mekanisme ini merujuk kepada proses pengemaskinian polisi *firewall* yang akan digunakan dalam kajian. Pendekatan ini nampak mudah namun, proses untuk mendapatkan masa bila ia akan digugurkan akan menjadi masalah jika melibatkan tahun, hari dan jam.

4.4 TRANSMISI DATA SISTEM *REDALERT*

Transmisi data dalam rangkaian sistem RedAlert menggunakan transmisi multicast. Penggunaan transmisi multicast memberi kelebihan kepada RedAlert supaya dapat memberi masa tindakan yang lebih pantas dan cepat. Namun penggunaan transmisi multicast terhad kepada satu subnet sahaja kerana ianya tidak dapat dilakukan untuk transmisi di luar dari rangkaian.

Penggunaan unicast sebagai alat ganti kepada multicast cuma digunakan pada dua atau lebih subnet dalam rangkaian dalam kiraan n-1, dengan n merujuk kepada bilangan subnet. Bermakna jika terdapat 3 subnet berbeza maka terdapat 2 agen unicast digunakan dengan satu merupakan subnet asas kepada RedAlert. Penggunaan unicast adalah bertujuan untuk proses penghantaran data dalam subnet berbeza.



Rajah 4.4 : Komunikasi multicast menggunakan unicast

Rangkaian multicast 1, 2 dan 3 masing-masing mempunyai konfigurasi berikut:

i. Semua hos mempunyai micro-*firewall* sendiri (iptables).

ii. Semua hos mempunyai RedAlertMulti. Agen multicast.

iii. Satu hos setiap rangkaian (subnet) mempunyai agen unicast (RedAlertUni).

iv. Satu hos mempunyai IDS, RedAlertIDS.

Transmisi data dihantar kepada semua hos dalam subnet yang sama dengan menggunakan komunikasi multicast. Ini bermakna semua akan menerima serentak apabila RedAlert menghantar polisi baru. Manakala bagi subnet yang berlainan pula, transmisi unicast dihantar kepada agen unicast pada setiap subnet. Peranan agen ini pula adalah membuat transmisi semula secara multicast dalam rangkaiannya sendiri. Proses transmisi ini adalah dalam masa nyata yang mana setiap pencerobohan berkaitan dikesan, polisi akan dibuat dan seterusnya dihantar kepada semua hos menggunakan kaedah multicast dan unicast.

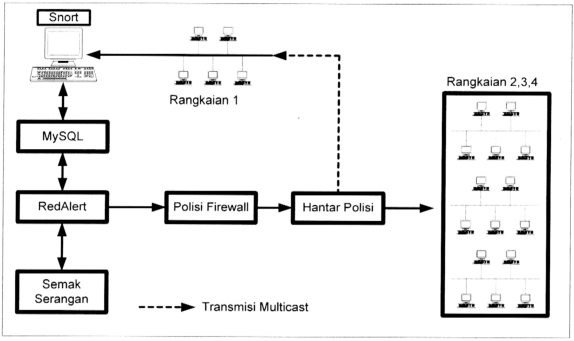
Transmisi yang digunakan dalam rangkaian sistem RedAlert adalah menggunakan kaedah penyulitan simetri. Bermakna semua agen melakukan proses penyulitan masa menghantar dan menyahsulit kembali apabila menerima. Kaedah penyulitan *XOR block* digunakan dan satu kunci boleh ubah digunakan mengikut keperluan pentadbir.

4.5 RANGKA KERJA SISTEM *REDALERT* SECARA KESELURUHAN

Integrasi sistem yang melibatkan gabungan beberapa sistem sedia ada dengan aplikasi baru membentuk satu sistem baru yang lebih baik, cepat dan memenuhi keperluan keselamatan. Sistem RedAlert merupakan gabungan lebih dari satu sistem termasuk aplikasi RedAlert yang terdiri daripada 4 aplikasi berbeza, sistem pengesan pencerobohan Snort dan iptables dalam Linux *firewall*. Pangkalan data yang digunakan adalah berasaskan MySQL yang merupakan pangkalan data sistem terbuka.

Secara keseluruhannya, gabungan aplikasi-aplikasi ini merupakan satu pembentukan yang baik dan merupakan mekanisme baru dalam menghasilkan tindakan dinamik dan automatik bagi aplikasi pengemaskinian polisi *firewall* berdasarkan tindakan dan proses sistem pengesan pencerobohan. Pengemaskinian dilakukan dengan menggunakan transmisi multicast yang memberi masa tindakan yang cepat dan pantas serta hampir sama dengan masa pencerobohan berlaku.

Gambaran mengenai proses rangka kerja secara keseluruhan digambarkan pada rajah 4.5 di sebelah. Gambaran ini merupakan gambaran awal dalam sistem *RedAlert*.



Rajah 4.5 : Sistem RedAlert secara keseluruhan

Dalam merangka proses kerja bagi sistem RedAlert ini, konfigurasi rangkaian dijangkakan adalah seperti berikut:

i. Setiap rangkaian 1, 2, 3, 4 disambung satu sama lain dengan rangkaian switch.

ii. Setiap hos dalam rangkaian 1, 2, 3, 4 menyokong aplikasi Multicast.

iii. Satu hos dari setiap rangkaian 1, 2, 3, 4 mengandungi agen unicast.

iv. Satu hos dari setiap rangkaian 1, 2, 3, 4 mempunyai Snort IDS, RedAlertIDS dan MySQL.

v. Setiap hos mempunyai micro-*firewall* sendiri. Iptables Linux *Firewall*.

Jadual 4.1 : Konfigurasi teknikal bagi setiap rangkaian

Rangkaian	Rangkaian 1	Rangkaian 2	Rangkaian 3	Rangkaian 4
Alamat IP Rangkaian	192.168.0.1/24	192.168.1.1/24	192.168.2.1/24	192.168.3.1/24
Gateway	192.168.0.254	192.168.1.254	192.168.2.254	192.168.3.254
Menyokong Multicast	Ya	Ya	Ya	Ya
Snort IDS	192.168.0.1	192.168.1.1	192.168.2.1	192.168.3.1
RedAlertIDS	192.168.0.1	192.168.1.1	192.168.2.1	192.168.3.1
MySQL	192.168.0.1	192.168.1.1	192.168.2.1	192.168.3.1
Agen Unicast	192.168.0.254	192.168.1.254	192.168.2.254	192.168.3.254
Sambungan Switch	Cisco Switch	Cisco Switch	Cisco Switch	Cisco Switch
Menyokong SPAN	Ya	Ya	Ya	Ya
Agen Multicast	192.168.0.1/24	192.168.1.1/24	192.168.2.1/24	192.168.3.1/24
Penceroboh	192.168.0.10	-	-	-
Mangsa	192.168.0.150	192.168.1.150	192.168.2.150	192.168.3.150
Micro-*firewall*	Iptables-1.2.7a	Iptables-1.2.7a	Iptables-1.2.7a	Iptables-1.2.7a
Operasi Sistem	RedHat 8.0	RedHat 8.0	RedHat 8.0	RedHat 8.0

Daripada konfigurasi di atas, rangka kerja sistem bagi RedAlert:

i. Penceroboh 1 dari dalam rangkaian 1 (192.168.0.10) , berusaha untuk menceroboh mangsa 1 (192.168.0.150) dalam rangkaian 1 menggunakan port sumber 1152 dan port destinasi 23 menggunakan aplikasi TELNET. Snort (192.168.0.1) mengesan penceroboh 1 dan menyimpan data berkaitan dalam MySQL.

ii. Data pencerobohan disimpan dalam MySQL (192.168.0.1) disemak oleh RedAlert dengan membanding pencerobohan yang perlu disekat dan pencerobohan yang tidak perlu disekat.

iii. Jika serangan memerlukan tindakan lanjutan, maka RedAlert akan memberi laluan kepada proses penghasilan polisi (192.168.0.1) berdasarkan data dalam pangkalan data. Polisi menghalang 192.168.0.10 melakukan serangan ke atas port 23 dibentuk.

iv. Selepas polisi dibentuk, polisi dihantar dari 192.168.0.1 ke semua rangkaian termasuk rangkaian 1, 2, 3 dan 4 dalam dua bentuk transmisi. Dalam rangkaian 1, subnet yang sama (192.168.0.1/24) transmisi multicast digunakan. Setiap hos akan menerima dalam masa yang sama.

v. Rangkaian 2,3, dan 4 akan menerima melalui transmisi unicast terlebih dahulu sebelum menyebarkan polisi melalui multicast. Penghantar 192.168.0.1 akan menghantar transmisi unicast kepada 3 hos unicast (192.168.1.254, 192.168.2.254, 192.168.3.254) tetapi tidak kepada 192.168.0.254.

vi. Penerima unicast ini kemudian akan menghantar maklumat kepada semua hos dalam rangkaian masing-masing.

vii. Penyerang dari 192.168.0.10 cuba menyerang tiga hos lagi dalam rangkaian yang lain selepas gagal menyerang hos 192.168.0.150. Mangsa seterusnya adalah 192.168.1.150, 192.168.2.150, 192.168.3.150. Penyerang gagal menyerang kerana polisi baru menghalang dari penceroboh menceroboh untuk kali kedua. Serangan gagal.

Rangka kerja sistem RedAlert melibatkan lebih daripada 1 rangkaian boleh melibatkan rangkaian yang lebih besar dan bukan dalam satu subnet yang sama. Cuma transmisi unicast perlu dilakukan bagi memastikan ianya dapat dilaksanakan. Penggunaan unicast sepenuhnya melambatkan proses pengemaskinian polisi. Jika menggunakan banyak hos dan banyak rangkaian proses akan menjadi lambat dan penyerang boleh menyerang sebelum polisi dikemaskini. Kelemahan penggunaan unicast ini diperbaharui dengan penggunaan multicast yang lebih baik dan pantas.

Proses pencerobohan yang berlaku pada gambaran di atas dapat dibezakan antara unicast dan multicast. Secara teorinya dinyatakan adalah seperti berikut:

Jadual 4.2 : Perbezaan transmisi multicast dan unicast

Transmisi	Multicast	Unicast	Multicast + Unicast
Bilangan Subnet	4	4	4
Bilangan Hos setiap subnet	200	200	200
Bilangan Hos Keseluruhan	800	800	800
Agen unicast	-	800	3
Masa transmisi multicast (saat)	1	1	1
Masa transmisi Unicast (saat)	1	1	1
Masa transmisi multicast menghantar polisi baru bagi setiap subnet (saat)	1	-	1
Masa transmisi unicast bagi setiap hos (saat)	-	800	3
Masa penghantaran kali ke-2 (saat)	-	-	3(1 saat bagi setiap subnet)
Bilangan hos yang dikemaskini	200	800	800
Proses kerja dijalankan	1	800	7
Sempurna proses pengemaskinian	Tidak	Ya	Ya
Kebarangkalian Serangan semula	Ya	Mungkin	Tidak
Masa proses selesai (saat)	1	800	7

Secara kiraan jelas menunjukkan bahawa transmisi multicast+unicast memberi jumlah masa proses yang rendah berbanding unicast sahaja. Walaupun masa transmisi multicast

rendah namun ianya masih tidak dapat menyempurnakan proses pengemaskinian polisi. Dengan demikian kebarangkalian serangan semula ke atas subnet yang lain tinggi berbanding kepada transmisi unicast dan gabungan multicast dan unicast. Kebarangkalian serangan ke atas hos lain dalam rangkaian lain adalah berkemungkinan jika menggunakan transmisi unicast. Ini kerana proses pengemaskinian polisi mengambil masa 800 saat bersamaan 13 minit untuk menyelesaikan semua proses. Walaupun penggunaan unicast sempurna tapi kebarangkalian serangan masih ada. Menggunakan gabungan multicast dan unicast juga berkemungkinan, jika serangan mengambil masa kurang dari 5 saat. Tetapi ini mungkin akan gagal apabila proses pengemaskinian selesai pada saat ke-7. Oleh itu penggunaan transmisi multicast amat baik dan tambahan unicast diperlukan jika transmisi memerlukan lebih daripada 2 subnet.