ANALISIS PERANCANGAN RULE PADA INTRUSION DETECTION DAN PREVENTION SYSTEM (IDPS) DALAM PENDETEKSIAN SERANGAN (ATTACKER) DI JARINGAN LAN

Hendri Mardani[1], Fazmah Arif Yulianto, ST.,MT.[2], Vera Suryani, S.T., MT.[3]

[1,2,3]Fakultas Informatika Institut Teknologi Telkom, Bandung
[1]**hendri.mardani@gmail.com**, [2]**fay@ittelkom.ac.id**, [3]**vra@ittelkom.ac.id**

Abstrak

Perkembangan internet yang semakin hari semakin meningkat baik dari sisi teknologi maupun penggunanya membawa dampak positif maupun negatif (cybercrime). Berdasarkan pernyataan tersebut maka dibutuhkanlah suatu bentuk pengamanan terhadap server khususnya. Salah satu caranya adalah dengan menggunakan Intrusion Detection and Prevention System (IDPS) dan yang akan diintegrasikan dengan firewall DMZ (Demilitarized Zone).

IDPS adalah proses memantau peristiwa yang terjadi dalam sistem komputer atau jaringan dan menganalisis tanda-tanda insiden yang mungkin terjadi seperti pelanggaran ataupun ancaman pelanggaran pada sistem computer dan jaringan. Pada tugas akhir ini akan dilakukan implementasi terhadap perancangan desain jaringan LAN menggunakan firewall DMZ yang diintegrasikan dengan IDPS yang menggunakan metode Signatured-Based Detection. Pada metode ini, tools yang telah digunakan berupa snort. Setiap serangan memiliki karakter yang berbeda dengan jenis serangan yang lain, maka diharapkan dengan menerapkan rul-rule baru dalam snort dapat mengantisipasi terjadinya pelanggaran terhadap sistem komputer dan jaringan komputer.

Dari hasil penelitian yang telah dilakukan maka topologi jaringan yang telah dibuat telah mampu menangkal setiap jenis serangan yang disesuaikan dengan skenario.

Kata kunci: Cybercrime, IDPS, DMZ, Computer Security, Firewall, Signatured-Based Detection.

Abstract

Developments of the internet that are getting increasing both in terms of technology and its users have a positive impact and negative (cybercrime). Based on these statements then required some form of security for server particulary. One solution is by using the Intrusion Detection and Prevention System (IDPS) and will be integrated with the DMZ (Demilitarized Zone) firewall.

IDPS is the process of monitoring the events in a computer system or network and analyzing the signs of an incident that may occur as a violation or threat on computer system and computer networks. in this thesis author has been successfully design an implementation of a LAN network using a firewall DMZ, integrated with IDPS using Signatured-Based Detection method. The tools that used in this method are snort and blockit. Each attach has a different character with another type, it is expected that by applying a new rule in snort can anticipate the occurrence of violations of the computer system and computer networks.

From the research that has been done, the network topology that has been made has been able to block any kind of attack which adjusted to the scenario.

Keywords: Cybercrime, IDPS, DMZ, Computer Security Policies, Firewall, Signatured-Based Detection.

1. Pendahuluan

Penggunaan dan kebutuhan akan teknologi informasi yang diaplikasikan dengan Internet dalam segala bidang seperti *e-education*, *e-banking*, *e-commerce*, dan *e-government* telah menjadi sesuatu yang biasa.Internet telah menciptakan dunia baru yang dinamakan *cyberspace* yaitu sebuah dunia komunikasi berbasis komputer yang menawarkan realitas yang baru berbentuk virtual (tidak langsung dan tidak nyata). Perkembangan Internet yang semakin hari semakin meningkat baik teknologi dan penggunaannya, serta membawa banyak dampak baik positif maupun negatif (*cybercrime*).

Berdasarkan pernyataan diatas maka diperlukanlah suatu bentuk pengamanan terhadap server khususnya yaitu dengan *Intrusion Detection Prevention System* (IDPS) [1][2][7]. IDPS adalah proses memantau peristiwa yang terjadi dalam sistem komputer atau jaringan dan menganalisis tanda-tanda insiden yang mungkin terjadi, seperti pelanggaran atau ancaman pelanggaran pada *komputer security policies*. Pada teknologi IDPS ada 3 metode untuk mendeteksi suatu *incident* (kejadian) yaitu *Signature-Based Detection*, *Anomaly-Based Detection*, dan *Stateful Protocol Analysis*[7]. *Cybercrime* dapat dihindari dengan membuat sistem keamanan yang menggunakan server salah satunya yaitu dengan menggunakan DMZ (*Demilitarized Zone*) pada jaringan LAN [3][4][5]. DMZ digunakan untuk melindungi jaringan internal dari serangan hacker sehingga tidak perlu membangun firewall di setiap server.

2. Landasan Teori

2.1. Intrusion Detection and Prevention Systems (IDPS)

Intrusion detection merupakan proses yang digunakan untuk memantau peristiwa yang terjadi didalam sistem komputer atau jaringan dan kemudian menganalisis tanda-tanda dari peristiwa tersebut, misalnya ancaman pelanggaran kebijakan keamanan komputer, penyerang untuk mendapatkan akses tidak sah ke sistem melalui internet, pengguna sistem yang menyalahgunakan hak-hak yang diberikan serta mencoba untuk mendapatkan hak tambahan oleh oknum-oknum yang tidak berwenang [1][2][12]. IDPS lebih difokuskan untuk identifikasi insiden yang mungkin terjadi. Misalnya, IDPS dapat mendeteksi ketika penyerang telah berhasil masuk ke dalam sistem dengan memanfaatkan kerentanan pada

sistem tersebut. IDPS kemudian bisa melaporkan kejadian tersebut kepada administrator jaringan, sehingga dengan cepat dapat dilakukan tindakan pencegahan untuk meminimalkan kerusakan yang disebabkan oleh insiden tersebut. IDPS juga dapat mencatat informasi yang dapat digunakan oleh penanganan insiden. Seperti contohnya, beberapa IDPS dapat dikonfigurasi dengan firewall yang memungkinkan untuk mengidentifikasi lalu lintas jaringan yang melanggar keamanan organisasi atau kebijakan yang tidak diperbolehkan. Beberapa IDPS juga dapat memantau transfer file dan mengidentifikasi orang yang mencurigakan, seperti menyalin database yang besar ke laptop pengguna yang tidak berhak. Ada banyak jenis teknologi IDPS, yang dibedakan berdasarkan peristiwa yang dapat dikenali dan metodologi yang digunakan untuk mengidentifikasi insiden. Disamping pemantauan dan menganalisis kejadian untuk mengidentifikasi aktifitas yang tidak diinginkan, semua jenis teknologi IDPS biasanya melakukan fungsi sebagai berikut :

a. *Identify Malicious Activity*
Merekam informasi yang berkaitan dengan peristiwa yang diamati, Informasi biasanya direkam secara lokal, dan juga mungkin akan dikirim ke sistem terpisah seperti server logging terpusat, informasi keamanan dan sistem manajemen perusahaan.

b. *Log Information (attempt to block/stop activity)*
Memberitahukan administrator keamanan terhadap peristiwa penting untuk diamati. Ini dikenal sebagai peringatan, terjadi melalui salah satu dari beberapa metode, termasuk e-mail, pesan pada user interface IDPS, pesan syslog, dan user-defined program dan script. Sebuah pesan pemberitahuan biasanya hanya mencakup informasi dasar tentang peristiwa, tugas dari administrator hanya perlu mengakses IDPS untuk informasi tambahan.

c. *Report Activity*
Membuat laporan merangkum peristiwa yang dimonitor dan memberikan rincian tentang peristiwa-peristiwa tertentu yang mencurigakan.

Pada IDPS ada 3 metode untuk mendeteksi suatu ancaman yaitu [1][12]:

1. Signature-Based Detection

Signature adalah sebuah pola yang sesuai dengan ancaman yang dikenal terlebih dahulu [1][12]. IDPS berbasiskan Signature memantau paket pada jaringan dan membandingkan dengan pola paket dengan signature sebelum konfigurasi dan sebelum ditentukan. Permasalahannya adalah adanya kemungkinan delay antara diketemukannya ancaman dan pengaplikasian signature pada IDPS untuk menemukan ancaman. *Signature-Based Detection* adalah proses yang membandingkan antara signature terhadap kejadian yang telah diamati untuk mengidentifikasi insiden yang mungkin terjadi.

2. Anomaly-Based Detection

Anomaly-Based Detection adalah proses untuk membandingkan definisi aktivitas yang dianggap normal terhadap kejadian yang sedang diamati untuk mengidentifikasi penyimpangan yang signifikan [1][12]. IDPS menggunakan *Anomaly-Based Detection* yang memiliki profil yang mewakili perilaku normal seperti hal-hal yang dilakukan oleh user, host, koneksi jaringan, atau aplikasi. Profil ini dikembangkan dengan memantau karakteristik dari aktivitas secara umum selama periode waktu tertentu. IDPS menggunakan metode statistik untuk membandingkan karakteristik dari kegiatan saat ini dengan ambang batas yang terkait dengan profil. Seperti mendeteksi ketika aktivitas browsing meningkat secara signifikan dengan bandwidth yang lebih dari yang diharapkan sehingga memberikan peringatan kepada admin mengenai anomali tersebut.

3. Stateful Protocol Analysis

Stateful Protocol Analysis merupakan proses membandingkan profil yang telah ditentukan dari definisi aktivitas protokol yang diketahui untuk setiap keadaan protocol yang umumnya diterima terhadap kejadian yang diamati untuk mengidentifikasi penyimpangan [1][12]. Berbeda dengan anomaly-based detection, dimana menggunakan profil tertentu dari sebuah host atau jaringan, *Stateful Protocol Analysis* bergantung pada profil umum yang dikembangkan oleh vendor yang menspesifikasikan bagaimana protokol tertentu harus dan tidak harus digunakan. Kata *"stateful"* dalam *Stateful Protocol Analysis* berarti bahwa IDPS mampu memahami dan melacak keadaan protocol jaringan, transportasi, dan aplikasi yang memiliki gagasan tentang keadaannya.

Komponen IDPS

Ada beberapa komponen dalam IDPS, diantaranya sebagai berikut [12] :

1. Sensor atau Agent
   Term sensor digunakan untuk memantau jaringan, termasuk network-based, wireless, dan teknologi network-behaviour. Sedangkan term agent digunakan untuk host berbasis teknologi IDPS.
2. Management Server
   Sebuah perangkat terpusat (centralized) yang menerima informasi dari sensor dan agent kemudian mengelola-nya apakah informasi tersebut bisa diidentifikasi. Prosesnya dengan melakukan pencocokkan informasi dari beberapa sensor atau agent,contohnya dalam menemukan *events triggered* yang melalui IP yang sama.
3. Database Server
   Adalah sebuah repository untuk merekam peristiwa (event) informasi yang dilakukan oleh sensor,agent, dan management server.
4. Console
   Adalah program yang menyediakan interface bagi pengguna IDPS dan administrator.

2.2. Demilitarized Zone (DMZ)

Demilitarized Zone (DMZ) adalah jaringan terpisah yang berdiri diluar jaringan yang diamankan [4][1][2]. User luar bisa mengakses DMZ, tapi tidak bisa memasuki jaringan yang kita amankan. Dengan menempatkan server-server yang memberikan layanan di dalam limit DMZ, maka akses dari luar hanya sebatas jaringan DMZ tersebut saja, user dari luar tidak akan mendapatkan akses terhadap jaringan diluar DMZ.

DMZ adalah subnetwork fisik dan logis yang berisi dan menyediakan dlayanan eksternal menuju sebuah jaringan yang lebih besar dan tidak terpercaya, biasanya Internet. Tujuan dari sebuah DMZ adalah untuk menambahkan sebuah lapisan keamanan tambahan untuk jaringan local area network (LAN) [11]. *Attacker* eksternal hanya mampu memiliki akses sampai kepada perangkat DMZ, tidak ke bagian internal.

Setiap layanan yang disediakan untuk pengguna pada jaringan eksternal dikelola pada DMZ. Layanan-layanan yang umum adalah :

a) Web servers
b) VoIP servers

2.3. SNORT

Snort merupakan suatu *network* IDS, *packet sniffer*, atau *packet logger*. Namun, yang lebih menarik adalah mengetahui Snort dari permulaannya daripada dari sekadar definisinya. Snort pada awalnya dibuat hanya sebagai *packet sniffer*. Pada November 1998, Marty Roesch membuat *packet sniffer* untuk Linux bernama APE.
Meskipun APE mempunyai fungsionalitas yang bagus, Roesch ingin *sniffer* yang juga bisa melakukan hal-hal sebagai berikut [1] :

1. Bekerja di berbagai sistem operasi.
2. Menggunakan hexdump payload dump (nantinya tcpdump mempunyai fungsionalitas ini).
3. Menampilkan semua paket di jaringan yang berbeda dengan cara yang sama (tcpdump tidak mempunyai fungsionalitas ini).

Tujuan Roesch awalnya membuat *sniffer* yang lebih baik untuk dirinya sendiri. Dia menciptakan Snort sebagai aplikasi *libcap*, yang membuatnya mudah berpindah antara menjadi pemfilter jaringan dan sebagai *sniffer*. Pada waktu itu, hanya tcpdump yang dikompilasi dengan *libcap*, sehingga sistem administrator bisa menggunakannya sebagai *sniffer*. Snort tersedia di Packet Storm (www.packetstormsecurity.com) pada 22 Desember 1998. Saat itu, Snort hanya terdiri dari 1600 baris kode dan memiliki total dua *file*. Hal ini sampai sekitar sebulan setelah awal pembuatan Snort dan Snort hanya digunakan sebagai *packet sniffer*. Roesch pertama kali menggunakan Snort untuk memonitor koneksi kabel modemnya dan men-debug aplikasi jaringan yang dia buat.

Signature-based analysis awal pada Snort (disebut rules-based analysis dalam komunitas Snort) muncul sebagai fungsionalitas di akhir Januari 1999. Hal ini merupakan langkah awal Snort menjadi pendeteksi serangan (intrusion detection) , dan memang saat itu Snort dapat dipakai sebagai *lightweight* IDS.

Koneksi VPN biasanya dibangun antara dua router akses internet yang dilengkapi dengan firewall dan perangkat lunak. Perangkat lunak ini harus diatur untuk menyambungkan ke mitra VPN, firewall harus dibentuk untuk memungkinkan pengaksesan, dan data yang dipertukarkan antara VPN harus dijamin dengan menggunakan enkripsi. Kunci enkripsi harus disediakan untuk semua mitra VPN sehingga pertukaran data hanya dapat dibaca oleh pihak yang berwenang.

2.4. RULE

Pemahaman tantang Rule adalah mutlak sangat penting untuk mendeteksi semua events sebagai false positif atau true positif, dan mampu menyempurnakan rule tersebut agar events yang akan datang bisa terus bermanfaat. Rule ini sangat penting dalam menjaga efektifitas dari snort.
Dengan menggunakan rule kita bisa mencocokkan hampir semua atribut dari sebuah paket. Beberapa yang lebih umum digunakan adalah port,size,Internet Protocol (IP) option,protocol,Internet Control Message Protocol (ICMP), dan Waktu. Secara umum yang paling penting adalah content dari sebuah packet atau payload. Kita bisa mencocokkan string sederhana, preprocessor normalized content, dan even kompleks regular expressions.
Rule tentu saja di design untuk mengambarkan dan mencocokkan keamanan terkait events dan serangan (attacks). Pada saat ini seperti kita ketahui didunia *networking* memiliki lingkungan interconnectivity yang lengkap, hampir semua sesuatu yang melintasi network dalam beberapa bentuk pada suatu waktu, dengan demikian snort sangat membantu dalam proses menemukan masalah yang akan terjadi.

1. Content

Kata kunci content adalah salah satu fitur yang lebih penting dari Snort. Hal ini memungkinkan pengguna untuk mengatur rule yang mencari content tertentu dalam payload paket dan tanggapan untuk memicu berdasarkan data tersebut.

Setiap kali pilihan untuk pencocokan pola content dilakukan, ketika memanggil Pola fungsi Boyer-Moore maka tes pencocokan dilakukan terhadap isi paket. Jika data persis cocok dengan argumen data string yang terdapat di mana saja dalam payload paket, tes ini berhasil dan sisanya dari tes pilihan aturan dilakukan. Perlu diketahui bahwa tes ini adalah case sensitive.

Data pilihan untuk kata kunci content adalah agak rumit, yang dapat berisi teks campuran dan data biner. Data biner umumnya tertutup dalam pipa ("|") karakter dan direpresentasikan sebagai bytecode. Bytecode merupakan data biner sebagai nomor heksadesimal dan merupakan metode ringkas (cepat) yang baik untuk mendeskripsikan data biner kompleks.

Berikut adalah contoh teks campuran dan data biner dalam aturan Snort [1].

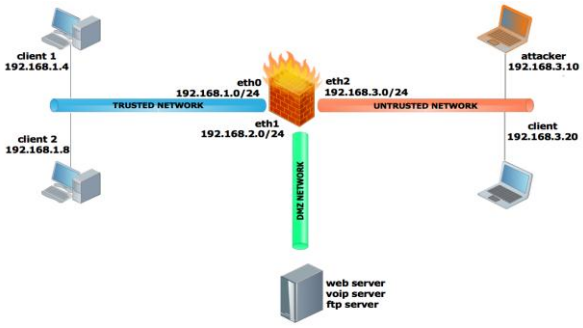
```
alert tcp any any -> 192.168.1.0/24 143 (content: "|90C8
C0FF FFFF|/bin/sh"; msg: "IMAP buffer overflow!";)
```

2. PCRE

PCRE merupakan pilihan pencocokan string yang sangat berguna. Snort menggunakan PCRE memungkinkan kita untuk melakukan beberapa pencocokkan yang sangat kompleks, dan pencocokan menggunakan PCRE ini dilakukan jika Opsi content tidak bisa dilakukan. Opsi PCRE (Perl-compatible regular expression) adalah bentuk lain dari pencocokan pola yang dapat digunakan untuk memeriksa isi dari paket [1]. Sesuai namanya, pilihan ini menggunakan ekspresi reguler untuk mencocokan paket.

3. Perancangan Dan Skenario Sistem

Pemodelan topologi jaringan dibawah ini (gambar 3.2) adalah perancangan dari Implementasi dan Analisis Keamanan Sistem pada Jaringan Lan dan DMZ menggunakan *Intrution Detection and Prevention System* (IDPS).



Gambar 3.2. Skenario Implementasi Web, VoIP, dan DMZ Server

Firewall yang digunakan adalah iptables pada ubuntu 10.04. Berikut adalah konfigurasi alamat jaringan yang akan digunakan :

- Ethernet 0 (trusted network)
  a) Mengizinkan beberapa akses layanan pada DMZ network
  b) Menutup semua akses dari untrusted network ke trusted network
- Ethernet 1 (DMZ network)
  a) Mengizinkan berlangsungnya akses layanan pada untrusted network
  b) Mengizinkan berlangsungnya akses layanan pada trusted network
- Ethernet 2 (untrusted Network)
  a) Mengizinkan berlangsungnya akses layanan pada DMZ network
  b) Menutup semua akses dari untrusted network ke trusted network

3.1 Tata Cara Penulisan Dan Konfigurasi Rule Snort

Secara umum letak konfigurasi snort pada linux terletak pada direktori */etc/snort.* Didalam direktori ini terdapat beberapa beberapa direktori lagi diantaranya direktori yang memuat tentang rule-rule snort.
Berikut adalah komponen rule snort :

a) *Rule Headers*
Rule Headers terdiri dari beberapa komponen diantaranya:
- *Action* yang berfungsi sebagai tindak lanjut ketika paket-paket jaringan terdeteksi oleh Snort. Contohnya *Alert* yang akan memberikan peringatan ketika paket-paket terdeteksi.
- *Protocol* merupakan jenis protokol yang digunakan dalam komunikasi jaringan. Contohnya protokol TCP, UDP dan IP.
- *Variable* yang berisi tentang kumpulan variable yang sudah didefinisikan pada konfigurasi snort. Misalnya *var HOME_NET 192.168.1.0/24*. Dengan adanya variabel ini, maka network 192.168.1.0/24 tidak perlu dituliskan secara lengkap ada rule snort. Cukup ditulis nama variabelnya saja.
- *Ip address* yang berisi daftar alamat yang berkomunikasi pada jaringan.
- *Port* yang berisi daftar port ketika terjadi komunikasi.
- *Direction* berisi tentang tanda petunjuk arah paket. Direction terdiri atas 2 tanda yaitu tanda '->' berarti pengecekan dilakukan dari source ip yang berada pada sebelah kiri tanda '>' ke destitation ip yang berada pada bagian kanan tanda tersebut. Yang kedua adalah tanda '<>' dimana ip address yang diberada pada bagian kiri bertidak sebagai source maupun destitation. Jika ip address tersebut bertindak sebagai source, maka ip address yang berada pada bagian kanan tanda '<>' akan bertindak sebagai destitation. Demikian pula sebaliknya.

Contoh dari *Rule Headers* adalah :

```
alert tcp $EXTERNAL_NET any ->
$HTTP_SERVERS $HTTP_PORTS
```

U

b)Rule Options
Rule Options terdiri dari beberapa komponen diantaranya :

- *Msg* yang berisi tentang judul rule dan pesan yang ditampilkan ketika serangan yang terjadi dideteksi oleh snort.
- *Content* berisi tentang isi paket-paket jaringan.
- *Flow* menunjukkan aliran paket data.
- *Uricontent* berisi tentang WEB URL dari paket-paket jaringan.
- *http_header* berisi tentang komponen http_header ketika terjadi komunikasi. Biasanya pada komunikasi HTTP.
- *Depth* berisi tentang spesifik paket yang akan diamatai.misalnya *depth:10* yang berarti bahwa pengecekan hanya dilakukan pada 10 byte paket awal.
- *Offsets* berisi tentang jumlah paket yang akan diabaikan ketika snort melakukan pengecekan. Misalnya *Offsets:100,* artinya snort 100 byte paket awal akan di abaikan oleh snort.
- *Classtype* yang berisi tentang kategor-kategori jenis serangan. Misalnya web-application-attack,trojan-activity.
- *Pcre* (*perl compatible regular expressions)* merupakan metode pencarian dan pendeteksian yang bisa mendeteksi paket-paket jaringan yang sifatnya dynamis.Misalnya *pcre:"/(stt|itt) telkom/i"* yang berfungsi untuk mencari kata "stt telkom" atau "itt telkom". Opsi 'I' digunakan untuk menyamakan pendeteksian karakter dengan huruf besar atau kecil.
- *Threshold* berisi tentang jumlah event yang akan ditampilkan oleh snort sesuai dengan waktu yang ditentukan. Hal ini digunakan untuk mengurangi event yang ditampilkan dalam jumlah banyak misalnya diakibatkan oleh proses *bruteforce.*
  Misalnya *threshold: type threshold, track by_dst, count 5, seconds 120***.**
  Konfigurasi ini berarti bahwa jika dalam waktu 120 detik ada 5 paket yang sama terdeteksi, maka snort akan memberikan event atau peringatan (alert).
- *Sid (snort Id)* merupakan ID yang harus diberikan untuk setiap rule yang sudah dibuat. Id rule 100 – 1.000.000 digunakan sendiri oleh group snort.org. Untuk user yang membuat rule sendiri disarankan menggunakan ID dari 1.000.001 – 1.999.999.
- *Rev* berisi tentang rule yang telah direvisi. Misalnya *Rev:3* yang berarti rule tersebut sudah mengalami modifikasi sebanyak 3 kali.

Contoh dari *Rule Options* adalah :

```
(msg:"WEB-IIS CodeRed v2 root.exe access";
flow:to_server,established;
uricontent:"/root.exe";nocase;
reference:url,www.cert.org/advisories/CA- 2001-
19.html; classtype:web-application-attack;
sid:1256; rev:8;)
```

3.2 Skenario Pengujian

Pada Tugas Akhir ini akan dilakukan pengujian layanan pada DMZ server dan client pada trusted network serta attacker pada untrusted network, melalui 4 skenario serangan dengan jenis-jenis serangan sebagai berikut :

1. Serangan menggunakan *trojan horse* atau *Remote access Tool* (RAT).
   Jenis *Trojan horse* yang akan digunakan dalam tugas akhir kali ini ada 5 jenis. Trojan ini sering digunakan oleh kalangan attacker didunia maya

untuk menyerang dan mencuri data targetnya.Contohnya Xtreme RAT, Cybergate RAT, Spy-Net RAT, Novaliyte RAT, LostDoor RAT.
Contoh rule yang dihasilkan antara lain :

alert tcp any any -> any any (msg: "BACKDOOR cybergate detected"; content:"|66 52 63 62 6B 50 63 7A 79 4F 74 62 59 50|"; nocase; sid:122213422; rev:2;)

2. Serangan menggunakan *trojan downloader* (web based trojan).
Trojan downloader sering digunakan oleh attacker sebagai aplikasi *backdoor* untuk mempertahankan akses terhadap sumber yang telah ia serang. Aplikasi ini juga sering digunakan untuk menjajalkan aplikasi berbahaya lainnnya tanpa sepengetahuan korban. Pada tugas akhir kali ini akan dianalisa 3 jenis *trojan loader* yang sering digunakan oleh attacker.
Contoh rule yang sudah dihasilkan antara lain :

alert tcp any any -> any any (msg: "koneksi elit loader terdeteksi"; content:"User|2d|Agent|3a| Mozilla/4.0"; http_header; content:"x|2d|type|3a| promake"; http_header; pcre:"/[a-z]+\.php\x3fv=\d+\x26id=./i";classtype:Trojan-activity; sid:11111112; rev:1;)*

3. Serangan sql injection terhadap website dengan database mysql.
Sql Injection merupakan serangan yang umum menimpa sebuah website. Hal ini terjadi karena kurangnya validasi terhadap inputan yang dimasukan oleh user. Pada tugas akhir ini, penulis memodifikasi dan menambahkan sebuah rule sql injection sehingga pendeteksian menjadi lebih akurat.
Contoh rule yang dihasilkan antara lain :

alert tcp any any -> any any (msg: "SQL Injection SELECT statement terdeteksi"; flow: established; pcre:"/(s|%73|%53)(e|%65|%45)(l|%6C|%4C)(e|%65|%45)(c|%63|%43)(t|%74|%45).(f|%66|%46)(r|%72|%52)(o|%6F|%4F)(m|%6D|%4D).*(\-\-|\/*|\#)/i"; sid: 2; rev: 3;)*

4. Serangan Password bruteforce yang menyerang server VOIP.
Contoh rule yang dihasilkan :

alert udp any any -> $HOME_NET 5060 (msg: "SIP account bruteforce terdeteksi"; content:"IP/2.0 401 Unauthorized"; threshold: type both, track by_src, count 10, seconds 10 priority:1; sid:12343; rev:1;)

4. Pengujian Dan Analisis

4.1. Skenario Serangan menggunakan Trojan Horse

4.1.1. Xtreme Remote Administration tools (RAT)

Dari hasil capture wireshark didapatkan hasil sebagai berikut :

```
    00000097  58 0d 0a                                          X..
    0000009A  d2 02 96 49 00 00 00 00  2c 00 00 00 00 00 00 00 ...I.... ,.......
    000000AA  78 01 4b 67 48 65 28 61  48 01 92 c5 0c d9 40 56 x.KgHe(a H.....@V
    000000BA  3e 43 01 10 16 01 f9 65  0c 99 40 b2 9c a1 86 c1 >C.....e ..@.....
    000000CA  88 c1 84 c1 00 48 03 00  d2 de 08 cb 70 69 6e 67 .....H.. ....ping
    000000DA  0d 0a                                          ..
00000249  d2 02 96 49 00 00 00 00  30 52 00 00 00 00 00 00 ...I.... 0R......
00000259  78 01 ec bb 75 58 54 df  f7 30 7a 68 51 42 10 94 x...uXT. .0zhQB..
00000269  06 09 29 69 a5 61 e8 94  ee 92 ee 90 90 16 06 44 ..)i.a.. .......D
```

Gambar 4.1.1.1 Capture paket data untuk serangan Xtreme RAT

```
0000020F  00 00 78 01 2b 67 48 65  48 62 48 66 48 64 c8 65 ..x.+gHe HbHfHd.e
0000021F  c8 61 c8 64 28 66 28 61  a8 61 50 62 08 65 08 66 .a.d(f(a .aPb.e.f
0000022F  70 62 50 60 08 03 8a a5  00 d5 e4 03 d9 2e 40 ba pbP`.... ......@.
0000023F  0c c8 4f 06 d2 4a 0c 00  45 69 0a 5d            ..O..J.. Ei.]
    00000095  58 0d 0a                                          X..
    00000098  d2 02 96 49 00 00 00 00  2c 00 00 00 00 00 00 00 ...I.... ,.......
    000000A8  78 01 4b 67 48 65 28 61  48 01 92 c5 0c d9 40 56 x.KgHe(a H.....@V
    000000B8  3e 43 01 10 16 01 f9 65  0c 99 40 b2 9c a1 86 c1 >C.....e ..@.....
    000000C8  88 c1 84 c1 00 48 03 00  d2 de 08 cb            .....H.. ....
```

Gambar 4.1.1.2 Capture paket data untuk serangan Xtreme RAT

Contoh rule yang dihasilkan berdasarkan string tersebut adalah :

alert tcp any any -> any any (msg: "Koneksi Xtreme RAT terdeteksi"; content:"|4b 67 48 65|(aH"; flow:established; pcre:"/\x4b\x67\x48\x65\x28/i";classtype:Trojan-activity; sid:111111; rev:1;)

Data yang diperoleh sebagai berikut :

```
    000000E2  70 69 6e 67 0d 0a                                ping..
000054FF  d2 02 96 49 00 00 00 00  2c 00 00 00 00 00 00 00 ...I.... ,.......
0000550F  78 01 2b 60 c8 67 c8 63  48 67 a8 61 30 61 30 64 x.+`.g.c Hg.a0a0d
0000551F  30 63 30 05 e2 1a 86 34  86 4c 86 1c 86 54 06 5d 0c0....4 .L...T.]
0000552F  86 22 86 44 86 12 06 00  87 3e 06 c7            .".D.... .>..
    000000E8  70 69 6e 67 0d 0a                                ping..
0000553B  d2 02 96 49 00 00 00 00  2a 00 00 00 00 00 00 00 ...I.... *.......
0000554B  78 01 2b 60 c8 67 c8 63  48 67 a8 61 30 63 30 04 x.+`.g.c Hg.a0c0.
0000555B  62 73 06 23 20 3b 8d 21  93 21 87 21 95 41 97 a1 bs.# ;.! .!.!.A..
0000556B  88 21 91 a1 84 01 00 87  52 06                  .!...... R.
```

Gambar 4.1.1.3 Trafik Xtreme Remote Administration tools (RAT)

Dan

```
00003B29  44 a6 bb 63                                          D..c
    000000D4  70 69 6e 67 0d 0a                                   ping..
00003B2D  d2 02 96 49 00 00 00 00  2a 00 00 00 00 00 00 00 ...I.... *.......
00003B3D  78 01 2b 60 c8 67 c8 63  48 67 a8 61 30 64 30 60 x.+`.g.c Hg.a0d0`
00003B4D  30 07 92 96 40 76 1a 43  26 43 0e 43 2a 83 2e 43 0...@v.C &C.C*..C
00003B5D  11 43 22 43 09 03 00 86  cc 06                  .C"C.... ..
```

Gambar 4.1.1.4 Trafik Xtream Remote Administration tools (RAT)

Dari data diatas bisa dilihat bahwa kode hexa **78 01 2b 60 c8 67 c8 63 48 67** memiliki nilai yang tetap. Kode **2b 60** bisa diganti dengan kode ascii sesuai dengan data yang ada pada bagian kanan menjadi +`. Rule yang dihasilkan dari signature teresebut adalah :

alert tcp any any -> any any (msg: "Xtreme RAT data akses terdeteksi"; content:"|78 01|+`|c8 67 c8 63 48 67|"; flow:established; priority:1; classtype:Trojan-activity; sid:1234332; rev:1;)

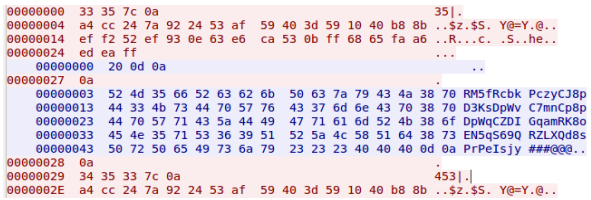
4.1.2. Cybergate Remote administration tools (RAT)

Berikut adalah beberapa contoh traffic yang dihasilkan selama simulasi:

```
00000000  33 31 7c 0a                                       31|.
00000004  a4 cc 24 7a 92 24 53 af  59 40 3d 59 10 40 b8 8b ..$z.$S. Y@=Y.@..
00000014  17 8a ac ea 96 09 fe d8  d6 66 a3 79 96 67 da    ........ .f.y.g.
    00000000  20 0d 0a                                       ..
00000023  0a                                              .
    00000003  52 4d 35 66 52 63 62 6b  50 63 7a 79 51 4d 35 6b RM5fRcbk PczyQM5k
    00000013  43 4a 38 70 56 33 34 70  43 4a 4b 76 44 5a 4b 70 CJ8pV34p CJKvDZKp
    00000023  43 4e 50 4b 44 62 44 72  49 71 66 76 52 4b 54 33 CNPKDbDr IqfvRKT3
    00000033  49 37 47 76 53 4a 35 66  45 34 72 49 54 72 62 6c I7GvSJ5f E4rITrbl
    00000043  53 74 6d 23 23 23 40 40  40 0d 0a              Stm###@@ @..
00000024  0a                                              .
00000025  34 34 36 7c 0a                                    446|.
0000002A  a4 cc 24 7a 92 24 53 af  59 40 3d 59 10 40 b8 8b ..$z.$S. Y@=Y.@..
```

Gambar 4.1.2.1 Trafik Cybergate Remote administration tools (RAT)

Dan



Gambar 4.1.2.2 Trafik Cybergate Remote administration tools (RAT)

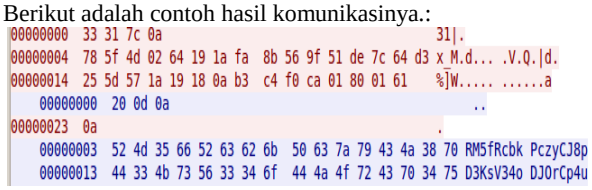
Rule yang dihasilkan dari signature tersebut adalah :

alert tcp any any -> any any (msg: "Koneksi TROJAN CYBERGATE terdeteksi"; content:"|a4 cc 24 7a 92 24 53 af 59 40 3d 59 10 40 b8 8b|"; flow:established; classtype:Trojan-activity; sid:111113; rev:1;)

alert tcp any any -> any any (msg: "Komunikasi TROJAN cybergate terdeteksi "; content:"pong"; flow:established; pcre:"/\x7c\w+\x23\x23\x23\d{5}.*\x7c/i"; classtype:Trojan-activity; sid:111222; rev:1;)

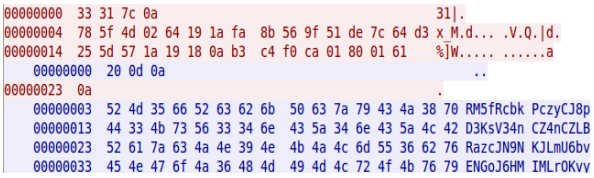
4.1.3. Spy-Net Remote Administration Tools (RAT)

Berikut adalah contoh hasil komunikasinya.:



Gambar 4.1.3.1 Spy-Net Remote Administration Tools (RAT)

Dan



Gambar 4.1.3.2 Spy-Net Remote Administration Tools (RAT)

Terlihat bahwa kode hexa **78 5f 4d 02 64 19 1a fa 8b 56 9f 51 de** memliki nilai yang tetap. Untuk itu kode ini bisa dijadikan sebagai acuan untuk pembuatan rule.

Rule yang dihasilkan dari signature tersebut adalah :

alert tcp any any -> any any (msg:"trojan spy-net terdeteksi"; content:"|78 5f 4d 02 64 19 1a fa 8b 56 9f 51 de|"; flow:established; classtype:Trojan-activity; sid:111116; rev:1;)

4.1.4. Lost-Door Remote Administration Tools (RAT)

Berikut adalah contoh data hasil capture menggunakan wireshark :

v1ct1mv1ct1m[\AS/]My Host[\AS/] Windows XP Professional[\AS/]ian-b530d4cae18[\AS/]4:01:35 PM[\AS/]United States[\AS/][\AS/]463.48 MB[\AS/]Yes[\AS/]Program Managerconnect

Dan

v1ct1mv1ct1m[\AS/]My Host[\AS/] Windows XP Professional[\AS/]ian-b530d4cae18[\AS/]7:56:21 AM[\AS/]United States[\AS/][\AS/]463.48 MB[\AS/]Yes[\AS/]rat

Rule yang dihasilkan sebagai berikut :

alert tcp any any -> any any (msg:"trojan lost-door v6 terdeteksi"; content:"v1ct1m"; nocase; content:"windows"; nocase; pcre:"/([\AS/])[a-zA-Z]+([\AS/]).*([\AS/])\w+([\AS/]).*/i"; classtype:Trojan-activity; sid:111118; rev:1;)

4.1.5. Novalite Remote Administration Tools (RAT)

Berikut adalah data yang didapat menggunakan wireshark:

MAININFO|412
MAININFO|ENU|192.168.1.8|ServerID|Dia @ IAN-B530D4CAE18|Windows XP Service Pack 3|1.0|
CONNECTED?
PING
PONG
CONNECTED?
CONNECTED?
PING

Rule yang dihasilkan adalah :

alert tcp any any -> any any (msg:"trojan novalite terdeteksi"; content:"MAININFO"; flow:established; pcre:"/MAININFO\x7c[A-Z]{3,}\x7c[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}.*\x7c(windows)/i"; classtype:Trojan-activity; sid:1111110; rev:1;)

4.2. Skenario Serangan menggunakan Trojan Downloader (web based trojan).

4.2.1. Trojan Gbot loader

Berikut adalah contoh hasil traffik data dari Gbot Loader:

GET //bot/gbot/getcmd.php?id=-1068335068&traff=0 HTTP/1.1
User-Agent: Opera/9.80 (Windows NT 5.1; U; ru) Presto/2.6.30 Version/10.63
Host: 192.168.3.11
Connection: close

HTTP/1.1 200 OK
Date: Mon, 25 Jun 2012 05:44:25 GMT
Server: Apache/2.2.17 (Unix) mod_ssl/2.2.17 OpenSSL/0.9.8r DAV/2 PHP/5.3.6
X-Powered-By: PHP/5.3.6
Content-Length: 44
Connection: close
Content-Type: text/html

Contoh rule yang dihasilkan adalah :

alert tcp any any -> any any (msg: "trojan gbot loader terdeteksi"; uricontent:"getcmd.php?id="; content:"User|2d|Agent|3a| Opera/9.80"; http_header; pcre:"/getcmd.php\?id\=\-\w+\x26traff\=.*/i";classtype:Trojan-activity; sid:1111122; rev:1;)

4.2.2. Trojan Elit Loader

Berikut adalah contoh hasil traffik data dari Elit Loader :

```
GET /bot/elit/doit.php?v=3&id=c0528024-76487-640-2394127-23295 HTTP/1.1
x-type: promake
User-Agent: Mozilla/4.0 (compatible; MSIE 6.1; Windows XP)
Host: 192.168.3.11

HTTP/1.1 200 OK
Date: Mon, 25 Jun 2012 06:06:50 GMT
Server: Apache/2.2.17 (Unix) mod_ssl/2.2.17 OpenSSL/0.9.8r DAV/2 PHP/5.3.6
X-Powered-By: PHP/5.3.6
Content-Length: 1
Content-Type: text/html; charset=windows-1251
```

Contoh Rule yang dihasilkan adalah :

```
alert tcp any any -> any any (msg: "koneksi elit loader terdeteksi"; content:"User|2d|Agent|3a| Mozilla/4.0"; http_header; content:"x|2d|type|3a| promake"; http_header; pcre:"/[a-z]+\.php\x3fv=\d+\x26id=.*/i";classtype:Trojan-activity; sid:11111112; rev:1; )
```

4.2.3. Trojan Pickpoket Loader

Berikut adalah contoh hasil traffik data dari PicketPoketLoader :

```
POST /bot/poket/proxy.php HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: umbra
Host: 192.168.3.11
Content-Length: 26
Cache-Control: no-cache

mode=1&UID=IAN-B530D4CAE18HTTP/1.1 200 OK
Date: Mon, 25 Jun 2012 07:13:58 GMT
Server: Apache/2.2.17 (Unix) mod_ssl/2.2.17 OpenSSL/0.9.8r DAV/2 PHP/5.3.6
X-Powered-By: PHP/5.3.6
Content-Length: 0
Content-Type: text/html
```

Contoh rule yang dihasilkan adalah :

```
alert tcp any any -> any any (msg: "koneksi pickpoket trojan detected"; uricontent:"proxy.php"; content:"User|2d|Agent|3a| umbra"; http_header; pcre:"/mode=\d{1,}\x26UID=\w+/i";classtype:Trojan-activity; sid:11111131; rev:1; )
```

4.3. Skenario Serangan Sql Injection

Berikut adalah contoh hasil traffik data dari Sql injection :

```
GET/sqli/events.php?id=-7104%20UNION%20ALL%20SELECT%20CONCAT%280x3a6676773a%2C0x4a4a47484e746f697565%2C0x3a6364683a%29%2C%20NULL%20UNION%20ALL%20SELECT%20CONCAT%280x3a6676773a%2C0x526f7a57524c6a56696b%2C0x3a6364683a%29%2C%20NULL%23 HTTP/1.1
Accept-Encoding: identity
Accept-Language: en-us,en;q=0.5
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
User-Agent: sqlmap/1.0-dev (r4766) (http://www.sqlmap.org)
Accept-Charset: ISO-8859-15,utf-8;q=0.7,*;q=0.7
Host: 192.168.2.11
Pragma: no-cache
Cache-Control: no-cache,no-store

HTTP/1.1 200 OK
Date: Mon, 25 Jun 2012 08:30:24 GMT
Server: Apache/2.2.17 (Unix) mod_ssl/2.2.17 OpenSSL/0.9.8r DAV/2 PHP/5.3.6
X-Powered-By: PHP/5.3.6
Connection: close
Content-Type: text/html
```

Dan

```
GET/sqli/events.php?id=-1+union+select+1,unhex(hex(concat(group_concat(username,0x3a,password))))%20from%20(select%20*%20from%20(select%20*%20from%20sqli.ms_user)t)t-- HTTP/1.1
Host: 192.168.2.11
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:10.0.2) Gecko/20100101 Firefox/10.0.2
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive

HTTP/1.1 200 OK
Date: Mon, 25 Jun 2012 08:30:31 GMT
Server: Apache/2.2.17 (Unix) mod_ssl/2.2.17 OpenSSL/0.9.8r DAV/2 PHP/5.3.6
X-Powered-By: PHP/5.3.6
Content-Length: 3025
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html
```

Dibawah ini adalah Rule yang dihasilkan apabila terjadi trafik serangan pada sql injection :

```
alert tcp any any -> any any (msg: "sql injection select terdeteksi"; uricontent:"union"; nocase; uricontent:"select"; nocase;uricontent:"select";nocase; uricontent:"information_schema"; nocase; flow: established;
pcre:"/(s|%73|%53)(e|%65|%45)(l|%6C|%4C)(e|%65|%45)(c|%63|%43)(t|%74|%45).*(f|%66|%46)(r|%72|%52)(o|%6F|%4F)(m|%6D|%4D).*(\-\-|\/\*|\#)/i";
priority:1; sid: 11111130; rev: 1;)
```

4.4. Skenario Serangan Password Bruteforce

Berikut adalah contoh hasil traffik data dari voip user bruteforce :

```
REGISTER sip:192.168.2.100 SIP/2.0
Via: SIP/2.0/UDP 127.0.0.1:5060;branch=z9hG4bK-2673280024;rport
Content-Length: 0
From: "1000"
```

<sip:1000@192.168.2.100>;tag=2772453338
Accept: application/sdp
User-Agent: friendly-scanner
To: "1000" <sip:1000@192.168.2.100>
Contact: sip:123@1.1.1.1
CSeq: 1 REGISTER
Call-ID: 2487472777
Max-Forwards: 70

SIP/2.0 401 Unauthorized
Via: SIP/2.0/UDP 127.0.0.1:5060;branch=z9hG4bK-2673280024;received=192.168.3.20;rport=5060
From: "1000" <sip:1000@192.168.2.100>;tag=2772453338
To: "1000" <sip:1000@192.168.2.100>;tag=as01452c5d
Call-ID: 2487472777
CSeq: 1 REGISTER
Server: FPBX-2.8.1(1.8.11.0)
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY, INFO, PUBLISH
Supported: replaces, timer
WWW-Authenticate: Digest algorithm=MD5, realm="asterisk", nonce="606a581a"
Content-Length: 0

Contoh rule yang dihasilkan adalah :

alert udp any any -> any (msg: "SIP account bruteforce terdeteksi"; content:"IP/2.0 401 Unauthorized"; threshold: type threshold, track by_src, count 10, seconds 10 priority:1; sid:12343; rev:1;)

5. KESIMPULAN

Berdasarkan hasil dari beberapa pengujian yang sudah diterapkan pada beberapa skenario yang sudah dibuat, dapat diambil kesimpulan bahwa :

1) Seluruh rule yang telah dibuat mampu mendeteksi serangan Trojan Horse, Trojan Downloader,Voip Password Bruteforce, dan Sql Injection.
2) Penggunaan kontent Statik berupa kode Hexa bisa digunakan sebagai acuan untuk pembuatan rule.
3) Kode PCRE yang digunakan pada rule sangat efektif untuk mendeteksi traffic data yang sifatnya dinamis.

Daftar Pustaka

[1] Baker A., Caswell B dan Poor Mike. Snort 2.1 Intrusion Detection 2nd Edition. Syngress Publishing, Inc. United States of America. 2004.

[2] Baker R.A. dan Esler J. Snort IDS and IPS Toolkit. Syngress Publishing, Inc. Canada. 2007.

[3] Clarke, Justin. Sql injection attacks and defense. Syngress Publishing, Inc. United States of America. 2009.

[4] Dubrawsky I. Designing and Building Enterprise DMZs. Syngress Publishing, Inc. Canada. 2006.

[5] Endler D. dan Mark C. Hacking Exposed VoIP: Voice Over IP Security Secrets & Solutions, McGraw-Hill/Osborne. United States of America. 2007.

[6] Engebretson, Patrick. The basics of hacking and penetration testing. Syngress Publishing, Inc. United States of America. 2011.

[7] Gregg,Michael. Build your own security Lab. Wiley Publishing,Inc. Indianapolis.2008.

[8] Harper A., Ness J., Lenkey G., Harris S., Eagle C., Williams T. Gray hat hacking third edition, McGraw. United States of America. 2011.

[9] Jajodia S., dan NoelAdvances S. Topological Vulnerability Analysis. George Mason University. USA. 2009.

[10] PARMAR,S.K. Information Resource Guide : Computer, Internet and Network Systems Security. Canada.

[11] Rash M. LINUX FIREWALLS Attack Detection and Response with iptables, psad, and fwsnort. No Starch Press, Inc. San Francisco. 2007.

[12] Scarfone K. dan Mell P. Guide to Intrusion Detection and Prevention Systems (IDPS). National Institute of Standards and Technology. USA. February 2007.