

EKSTRAKSI CIRI SIGNIFIKAN PADA ANALISA FORENSIK JARINGAN MENGUNAKAN TEKNIK INTELLIGENSIA BUATAN

**Tugas Kuliah
EC 7010
Keamanan Sistem Lanjut**

**Oleh :
Suryani Alifah
23202071**



**Magister Teknologi Informasi-Teknik Elektro
Institut Teknologi Bandung
2004**

DAFTAR ISI

ABSTRAK	2
BAB I PENDAHULUAN	3
BAB II FORENSIK JARINGAN (<i>NETWORK FORENSIC</i>)	4
2.1 Definisi Forensik Jaringan.....	4
2.2 Sistem <i>Offline vs Online</i>	4
2.3 Data Volatil vs Non-Volatil.....	4
2.4 Proses Forensik Jaringan.....	5
2.4.1 Akuisisi dan Pengintaian (<i>Reconnaissance</i>).....	5
2.4.1.1 Pengumpulan Data Volatil.....	6
2.4.1.2 Melakukan <i>Trap</i> dan <i>Trace</i>	7
2.4.2. Analisa Data.....	9
2.4.2.1 <i>Log File</i> sebagai Sumber Informasi.....	9
2.4.2.2 Interpretasi Trafik Jaringan.....	9
2.4.2.3 Pembuatan <i>Time Line</i>	11
2.5. <i>Tool</i> dan <i>Toolkit</i> untuk Forensik Jaringan.....	13
2.5.1 <i>The Coroner's Toolkit (TCT)</i>	13
2.5.2 <i>TCTUtils</i>	14
2.5.3 <i>Autopsy Forensic Browser</i>	14
BAB III PENGELOMPOKAN TINGKAT SIGNIFIKANSI INPUT	15
4.1. Metoda Pengelompokan Berdasarkan Kinerja	15
4.1.1 Metrik Kinerja untuk Pengelompokan Berbasis <i>Support Vector Machine (SVM)</i>	15
4.1.2 Metrik Kinerja untuk Pengelompokan Berbasis <i>Artificial Neural Network (ANN)</i>	16
4.2. Metoda Pengelompokan Berdasarkan <i>Support Vector Decision Function (SVDF)</i>	16
4.3. Perbandingan Ekstraksi <i>Significant Feature</i> Menggunakan <i>SVM</i> vs <i>ANN</i>	17
BAB IV KESIMPULAN	21
DAFTAR PUSTAKA	22

ABSTRAK

Forensik jaringan (network forensic) merupakan proses menangkap, mencatat dan menganalisa aktivitas jaringan guna menemukan bukti digital (*digital evidence*) dari suatu serangan atau kejahatan yang dilakukan terhadap , atau dijalankan menggunakan, jaringan komputer sehingga pelaku kejahatan dapat dituntut sesuai hukum yang berlaku.

Volume data yang diperoleh pada proses forensik tersebut sangatlah besar sehingga diperlukan ekstraksi ciri-ciri yang signifikan untuk meningkatkan ketelitian dan menghemat waktu analisa. Teknik intelligensia buatan dapat digunakan untuk melakukan otomatisasi ekstraksi ciri signifikan pada analisa forensik jaringan.

Pada paper ini dikemukakan dua buah teknik intelligensia buatan untuk proses ekstraksi ciri signifikan tersebut pada analisa forensik jaringan secara *offline*, yakni *Support Vector Machine (SVM)* dan *Artificial Neural Network (ANN)*. Pada konteks tersebut *SVM* mempunyai beberapa kelebihan dibandingkan dengan *ANN* yakni tingkat ketelitian deteksi yang lebih tinggi, skalabilitas yang lebih baik serta *training time* dan *running time* yang lebih cepat.

BAB I

PENDAHULUAN

Kemajuan teknologi informasi telah memicu perkembangan teknologi jaringan data, baik lokal maupun global, di mana banyak pihak memanfaatkan untuk berbagai aplikasi, diantaranya dunia bisnis (*e-commerce*), perbankan (*e-banking*), pendidikan (*e-learning*), militer, kepolisian, pemerintahan (*e-government*), dll. Namun dibalik manfaat yang sangat besar tersebut muncul masalah baru yakni kejahatan dunia maya (*cyber criminal*) akibat adanya lubang keamanan sebagai dampak sistem yang bersifat terbuka.

Kejahatan dunia maya, yang merupakan kejahatan dengan menggunakan teknologi informasi sebagai instrumen atau sarannya, telah mendorong lahirnya forensik jaringan (*network forensic*) sebagai jawaban atas maraknya kasus tersebut. Meningkatkan kualitas tool dan teknik untuk analisa forensik jaringan sangatlah diperlukan karena kita harus berlomba dengan penjahat dunia maya yang semakin lama semakin canggih. Salah satu teknik yang cukup mendesak untuk ditingkatkan adalah otomatisasi ekstraksi ciri signifikan pada analisa forensik mengingat sangat banyaknya data mentah yang harus dianalisa. Ekstraksi ciri untuk membuang data-data yang tidak berguna bagi analisa maupun mengelompokkan data mana terpenting, data mana yang kurang penting akan mempercepat proses analisa forensik dan meningkatkan akurasi sehingga meningkatkan kinerja secara keseluruhan.

Salah satu analisa yang mengolah data yang sangat besar adalah analisa *log*, karena mencatat seluruh aktifitas yang dilaksanakan pada jaringan. Selanjutnya *log file* harus diubah ke dalam format yang sesuai untuk dapat dibandingkan dengan pola serangan yang dikenali, penyimpangan dari perilaku normal jaringan ataupun penyimpangan dari kebijakan keamanan yang diterapkan pada jaringan. Log file setelah beberapa waktu kemudian akan diperbaharui sehingga proses di atas harus terus diulang. Dengan demikian tanpa teknik dan tool yang dapat mengekstraksi data yang signifikan, akan dibutuhkan *resources* yang sangat besar sehingga proses forensik jaringan menjadi tidak efektif.

Support Vector Machine (SVM) dan *Artificial Neural Network (ANN)* adalah teknik intelligensia buatan yang dapat digunakan untuk membangun sistem forensik jaringan. Kedua mesin pembelajaran tersebut dapat digunakan untuk mengekstraksi ciri-ciri penting pada analisa forensik jaringan *offline*. Keduanya mengelompokkan data berdasarkan tingkat signifikansi dengan menggunakan metrik kinerja masing-masing. *SVM* mempunyai beberapa kelebihan dibandingkan dengan *ANN* sehubungan dengan ekstraksi ciri tersebut [1].

BAB II

FORENSIK JARINGAN

2.1. Definisi Forensik Jaringan

Forensik jaringan (*Network forensic*) merupakan proses menangkap, mencatat dan menganalisa aktivitas jaringan guna menemukan bukti digital (*digital evidence*) dari suatu serangan atau kejahatan yang dilakukan terhadap , atau dijalankan menggunakan, jaringan komputer sehingga pelaku kejahatan dapat dituntut sesuai hukum yang berlaku.

Bukti digital dapat diidentifikasi dari pola serangan yang dikenali, penyimpangan dari perilaku normal jaringan ataupun penyimpangan dari kebijakan keamanan yang diterapkan pada jaringan.

2.2. Sistem *Offline* vs *Online*

Sistem *offline* merupakan sistem yang telah diisolasi dari jaringan, sedangkan sistem *online* merupakan sistem yang masih terhubung ke jaringan. Analisa pada sistem *offline* tersebut relatif lebih sederhana dibandingkan analisa pada sistem *online* karena *hard disk* dapat dijaga *read only* dan *copy*/salinan *bit-stream* dapat dibuat bersamaan dengan *cryptographic hash* dari sistem aslinya. *Hash* tersebut kemudian dapat dibandingkan dengan *hash* dari salinan *bit stream* untuk menunjukkan integritas salinannya.

Namun demikian pada beberapa keadaan, susah bahkan tidak mungkin untuk mengisolasi sistem dari jaringan, sehingga analisa harus dilakukan pada sistem *online*. Kesukaran terjadi karena sistem *online* masih terhubung ke jaringan, sehingga sistem terus berubah walaupun tanpa intervensi analisis forensik sekalipun.

2.3. Data Volatil vs Non-Volatil

Berdasarkan sifatnya, informasi terbagi menjadi dua jenis, yakni :

(1) Data volatil

Merupakan data sistem yang bersifat transien sehingga harus sesegera mungkin diambil /dicatat setelah terjadi insiden. Data ini bersumber dari keadaan hubungan jaringan, tabel *router*, tabel proses, *open port* dan *user*. Untuk menangkap data ini diperlukan *tool* khusus.

Data volatil terdiri dari :

- Data jaringan
Yakni komunikasi aktual antara sistem sasaran dengan sistem lainnya
- Daftar proses aktif

Berisi daftar program atau *daemon* yang aktif pada sistem sasaran

- Daftar *logged-in user*

Berisi daftar pengguna pada sistem sasaran

- *Open file*

File (hidden) atau *Trojan (rootkit)* apa yang dimasukkan ke sistem sasaran

(2) Informasi non-volatil

Merupakan informasi yang masih ada setelah proses *reboot*, yang terdiri dari :

- *Configuration setting*
- File data dan sistem
- *Registry setting*

2.4. Proses Forensik Jaringan

Proses forensik jaringan terdiri dari beberapa tahap, yakni :

1) Akuisisi dan pengintaian (*reconnaissance*)

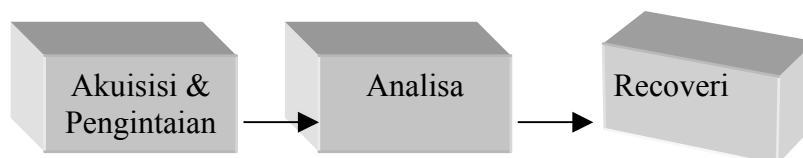
Yaitu proses untuk mendapatkan/mengumpulkan data volatil (jika bekerja pada sistem *online*) dan data non-volatil (disk terkait) dengan menggunakan berbagai *tool*.

2) Analisa

Yaitu proses menganalisa data yang diperoleh dari proses sebelumnya, meliputi analisa *real-time* dari data volatil, analisa *log-file*, korelasi data dari berbagai divais pada jaringan yang dilalui serangan dan pembuatan *time-lining* dari informasi yang diperoleh.

3) Recoveri

Yaitu proses untuk mendapatkan/memulihkan kembali data yang telah hilang akibat adanya intrusi, khususnya informasi pada disk yang berupa file atau direktori.



Gambar 1. Proses forensik

2.4.1. Akuisisi dan Pengintaian (*Reconnaissance*)

Tahap awal proses forensik merupakan hal yang kritis karena menentukan keberhasilan proses forensik. Tahap ini merupakan proses pengumpulan data dan pengintaian.

2.4.1.1. Pengumpulan Data Volatil

Data volatil dikumpulkan dari berbagai sumber, yakni register proses, memori virtual dan fisik, proses-proses yang sedang berjalan, maupun keadaan jaringan. Sumber informasi tersebut pada umumnya mempunyai informasi dalam periode yang singkat, sehingga waktu pengumpulan bersifat kritis dan harus sesegera mungkin diambil setelah terjadi insiden. Misalnya alamat *MAC* (*Media Access Control*) dari computer yang berkomunikasi dengan computer sasaran yang berada pada subnet yang sama, yang tersimpan pada *ARP* (*Address Resolution Protocol*) *cache*, segera dibuang setelah terjalin komunikasi dengan computer lainnya, sehingga data tersebut harus segera diambil.

Sumber informasi volatil yang penting beserta instruksi-instruksi yang digunakan untuk menangkap informasi tersebut diantaranya :

- Proses-proses yang sedang berjalan (*ps* atau */proc*)
- Hubungan jaringan yang aktif (*netstat*)
- *ARP cache* (*arp*)
- *List of open file* (*lsop*)
- Memori fisik dan virtual (*/dev/mem*, */dev/kmem*)

Instruksi */sbin/arp -v* menampilkan isi dari *ARP cache*. Contoh berikut menggambarkan bahwa *ARP cache* pada computer ini mempunyai 3 alamat *MAC* yang mempunyai label **Hwaddress**.

```
[jpc@whammo jpc]$ /sbin/arp -v
Address HWtype HWaddress
192.168.1.105 ether 00:10:xx:xx:xx:xx
192.168.1.104 ether 00:08:xx:xx:xx:xx
192.168.1.1 ether 00:06:xx:xx:xx:xx
```

Instruksi *netstat* menampilkan hubungan jaringan dan *listening port*. Berikut adalah contoh hasil menjalankan instruksi *netstat* :

```
[jpc@whammo jpc]$ netstat -an
Active Internet connections (servers and
established)
Proto LocalAddr ForeignAddr State
tcp 0.0.0.0:31337 0.0.0.0:* LISTEN
tcp 0.0.0.0:143 0.0.0.0:* LISTEN
tcp 0.0.0.0:119 0.0.0.0:* LISTEN
tcp 0.0.0.0:25 0.0.0.0:* LISTEN
tcp 0.0.0.0:12346 0.0.0.0:* LISTEN
tcp 192.168.1.100:22 92.168.1.105:32834
ESTABLISHED
```

... [output truncated]

Terdapat hubungan tunggal yang dibangun pada *port* 22 (*secure shell*, SSH) dan lima *listening port*, termasuk *imap* (143), *nntp* (119) dan *smtp* (25) yang merupakan layanan normal yang diharapkan pada server, sedangkan dua *listening port* lainnya adalah 31337 (*port* yang biasanya digunakan oleh *Trojan Back Orifice*) dan 12346 (*port* yang digunakan oleh *Trojan NetBus*).

Jika suatu sistem terhubung ke jaringan maka informasi volatil dapat dikumpulkan dan dikirim ke sistem forensik menggunakan tool *netcat*.

```
[jpc@whammo jpc]$ (ps -aux; netstat -  
an; lsop) | nc 192.168.1.104 10009
```

Perintah ini mengumpulkan informasi proses-proses yang sedang berjalan, (*ps-aux*), daftar *open file* (*lsop*) dan mengirimkannya melalui hubungan jaringan ke mesin forensik lokal menggunakan *netcat* (*nc*) melalui *port* 10009.

2.4.1.2. Melakukan *Trap* dan *Trace*

Trap dan *trace* merupakan proses untuk memonitor *header* dari trafik internet tanpa memonitor isinya (tidaklah legal untuk memonitor isi dari suatu komunikasi data). Proses ini merupakan cara non intrusif untuk menentukan sumber serangan jaringan atau untuk mendeteksi kelainan trafik karena hanya mengumpulkan *header* paket TCP/IP dan bukan isinya.

Contoh berikut menggambarkan bagaimana kemampuan *tcpdump* menampilkan hasil simulasi *trap* dan *trace* :

```
[root@whammo jpc]# /usr/sbin/tcpdump  
tcpdump: listening on eth0  
[1] 14:36:02.232495  
whammo.cobalt.net.32836 > 207-171-  
182-16.amazon.com.http: S 4042  
034720:4042034720(0) win 5840 <mss  
1460> (DF)  
[2] 14:36:02.248120  
whammo.om.cobalt.net.32791 >  
nsl.cobalt.net.domain: 38345+ PTR?  
16 .182.171.207.in-addr.arpa. (45)  
(DF)  
[3] 14:36:02.313807 207-171-182-  
16.amazon.com.http >  
whammo.cobalt.net.32836: S  
12410:12410(0) ack 4042034721 win
```

```

8760 <mss 1460,eol> (DF)
[4]14:36:02.313854
whammo.cobalt.net.32836 > 207-171-
182-16.amazon.com.http: . Ack 1 win
5840 (DF)

```

Paket[1], [3] dan [4] berisi waktu paket diterima, *port* dan alamat sumber, *port* dan alamat tujuan, *TCP flag*, nomor urut, dll. Empat deret paket menggambarkan tiga cara *handshake* (*SYN*, *SYN-ACK*, *ACK*) antara sebuah computer dan *Amazon.com*, dengan resolusi *DNS* berada pada tahap [2]. Tidak terdapat waktu ketika paket dikumpulkan atau ditampilkan. Paket *default* yang ditangkap untuk *tcpdump* adalah 68 *byte*. Jumlah total *IP header*, minimum 20 *byte* dan *TCP header* juga sama, sehingga total berjumlah 40 *byte*. Contoh di atas menggunakan panjang *default* 68 *byte*.

Trap dan *trace* dapat digunakan oleh analis forensik untuk menjawab beberapa pertanyaan kritis , yakni :

- Apakah alamat IP sumber mencurigakan?
- Apakah alamat IP dan/ atau nomor *port* tujuan mencurigakan?
Misal beberapa *port* yang sangat dikenal digunakan oleh *Trojan* adalah 31337 untuk *Back Orifice* dan 12345 untuk *NetBus*.
- Apakah terdapat fragmentasi yang aneh?
Fragmentasi sering digunakan untuk membingungkan IDS dan *firewall*.
- Apakah *TCP flag* mencurigakan?
Misal, beberapa *flag* tidak pernah terjadi bersama-sama, seperti R & F (*reset & fin*), *F alone*, dll. Penyerang menggunakan teknik ini untuk menentukan sistem operasi dari computer sasaran.
- Apakah ukuran paket mencurigakan?
Paket *SYN* awal seharusnya membawa data 0 *byte*.
- Apakah tujuan *port* merupakan layanan yang valid?
Layanan yang valid biasanya ditampilkan dalam dalam file */etc/services* pada mesin Linux.
- Apakah trafik mengikuti standar *RFC*?
- Apakah *timestamp* trafik?

2.4.2. Analisa Data

2.4.2.1. Log File sebagai Sumber Informasi

Keberhasilan proses forensik sangat ditentukan oleh kualitas dan kuantitas informasi yang terkumpul. *Log file* dapat merupakan sumber informasi yang penting bagi proses forensik. *Log file* mengandung informasi tentang berbagai sumber daya sistem, proses-proses dan aktivitas pengguna. *Protocol analyzer, sniffer, server SMTP, DHCP, FTP dan WWW, router, firewall* dan hampir semua aktivitas sistem atau user dapat dikumpulkan dalam *log file*. Tetapi jika administrator sistem tidak dapat mencatat, maka fakta yang diperlukan untuk menghubungkan pelaku dengan insiden tidak ada. Sayangnya penyerang dan penjahat yang pintar mengetahui hal ini dan tujuan pertamanya adalah merusak atau mengubah *log file* untuk menyembunyikan aktivitas mereka. Hal kedua yang penting tetapi sering dilupakan adalah sistem *clock*. Pencatatan suatu file berhubungan dengan *time stamp* dan *date stamp* yang memungkinkan analisis forensik untuk menentukan urutan kejadian. Tetapi jika sistem *clock* tidak dikoreksi/dikalibrasi secara berkala dapat dimatikan dari mana saja dari beberapa detik sampai beberapa jam. Hal ini menyebabkan masalah karena korelasi antara *log file* dari computer yang berbeda yang mempunyai sistem *clock* yang berbeda akan menyulitkan bahkan tidak mungkin mengkorelasikan kejadian. Solusi yang sederhana untuk mensinkronisasi *clock* adalah seluruh server dan sistem berjalan pada suatu *daemon* seperti *UNIX ntpd daemon*, yang mensinkronisasi waktu dan tanggal sistem secara berkala dengan suatu *atomic clock* yang disponsori pemerintah.

2.4.2.2. Interpretasi Trafik Jaringan

Untuk dapat mengidentifikasi trafik jaringan yang tidak normal dan mencurigakan, harus dapat mengenali dengan baik pola trafik jaringan yang normal.

Contoh berikut mengidentifikasi sederetan paket jaringan:

```
[1] 10:51:18.097489
whammo.cobalt.net.33593 >
server1.unomaha.edu.ftp: S
4022300050:4022300050(0) win 58 40
<mss 1460> (DF)
[2] 10:51:18.115166
server1.unomaha.edu.ftp >
whammo.cobalt.net.33593: S
1161910619:1161910619(0) ack 40
22300051 win 17520 <mss 1460> (DF)
[3] 10:51:18.115218
whammo.cobalt.net.33593 >
server1.unomaha.edu.ftp: . ack 1 win
5840 (DF)
```

Deretan paket di atas merupakan *TCP handshake* yang khas (*SYN, SYN-ACK, ACK*) sehingga merupakan trafik yang normal. Contoh tersebut menggambarkan usaha yang normal untuk terhubung ke *port 21*, ftp atau suatu server.

Contoh berikut ini menggambarkan hal yang berbeda :

```
[1] 10:51:18.097489
192.168.100.105.33593 >
server1.unomaha.edu.netbios-ssn: S
4022300050:4022300050(0) win 58 40
<mss 1460> (DF)
[2] 10:51:18.115218
192.168.100.104.33594 >
server1.unomaha.edu.netbios-ssn: S
4022300051:4022300051(0) win 58 40
<mss 1460> (DF)
[3] 10:51:18.097489
192.168.100.107.33595 >
server1.unomaha.edu.netbios.ssn: S
4022300052:4022300052(0) win 58 40
<mss 1460> (DF)
[4] 10:51:18.115218
192.168.100.105.33596 >
server1.unomaha.edu.netbios-ssn: S
4022300053:4022300053(0) win 58 40
<mss 1460> (DF)
[5] 10:51:18.097489
192.168.100.111.33597 >
server1.unomaha.edu.netbios-ssn: S
4022300054:4022300054(0) win 58 40
<mss 1460> (DF)
```

Pada contoh di atas, meskipun terlihat terdapat beberapa komputer yang berbeda yang berusaha untuk memulai *three-way handshake*, terdapat beberapa petunjuk yang menunjukkan bahwa trafik tersebut abnormal dan juga suatu serangan ke server1.unomaha.edu.

Pertama, alamat IP sumber terlihat tidak lazim (palsu) karena berupa satu set alamat IP cadangan yang biasanya digunakan di dalam jaringan sebagai alamat privat (misalnya dengan NAT) dan tidak pernah muncul di internet. Juga *time-stamp* terlalu berdekatan, dan port sumber dan nomor urut yang naik secara seragam merupakan petunjuk bahwa hal ini merupakan paket yang tidak normal.

Paket ini menjadi *SYNflood*, suatu jenis DoS (*denial of service*), suatu serangan terhadap server ini menggunakan *port 139 (NETbios)*.

2.4.2.3. Pembuatan *Time Lining*

MAC (Modified Access Creation) time merupakan tool yang sangat berguna untuk menentukan perubahan file, yang dapat digunakan untuk membuat *time lining* dari kejadian-kejadian.

M-times berisi informasi tentang kapan file dimodifikasi terakhir kali, *A-times* mengandung informasi waktu akses terakhir (membaca atau mengeksekusi) dan *C-times* berisi waktu terakhir status file diubah.

Berikut adalah contoh penggunaan *MAC times* untuk membuat *time line* guna mengetahui apa saja file yang telah dibuat dalam */var/log* directory sejak 9/6/2002 pada komputer tertentu :

- Mula-mula *mac-robber* dan *mactime* dijalankan dari *TASK forensic toolkit* pada direktori tersebut, dan hasilnya sbb :

```
[root@whammo /var] mac-robber /log >
body.mac
[root@whammo /var] mactime -b
body.mac 9/1/2002 > timeline
```

```
Tue Sep 10 2002 15:21:34
63344 ..c -rw-r--r-- 0 0 1418328
log/ksyms.3
63344 mac -rw-r--r-- 0 0 1418346
log/ksyms.0
63344 ..c -rw-r--r-- 0 0 1418334
log/ksyms.1
```

```
Tue Sep 10 2002 15:21:53
2941 m.c -rw----- 0 0 1417388
log/secure
16714 m.c -rw----- 0 0 1417389
log/maillog
```

```
Tue Sep 10 2002 15:21:54
9242 m.c -rw----- 0 0 1418347
log/boot.log
```

```
Tue Sep 10 2002 15:21:57
31164 m.c -rw-r--r-- 0 0 1418332
log/XFree86.0.log
```

```
Tue Sep 10 2002 15:22:06
19136220 m.c -rw-r--r-- 0 0 1417061
log/lastlog
```

```
Tue Sep 10 2002 16:26:26
```

```
81408 m.c -rw-rw-r-- 0 22 1418356
log/wtmp
```

Setelah tanggal, waktu dan ukuran file, terdapat tiga digit *field* yang berisi indikasi waktu terkait, apakah *M-times*, *A- times*, *C-times* atau kombinasinya. Perubahan *MAC* diorganisasikan dengan tanggal dan waktu. Dari daftar tersebut seseorang dapat menentukan apakah file berada pada direktori tersebut, waktu file dibuat, waktu terakhir dimodifikasi dan waktu terakhir diakses.

Ketika file dihapus pada suatu operating sistem, maka yang dihapus hanyalah pointer ke file yang menunjukkan bahwa ruang memori yang digunakan oleh file tersebut bebas untuk digunakan lagi, sedangkan isi file tidak terhapus. *TASK* atau *TCT* dapat digunakan untuk mengidentifikasi file yang terhapus, memperoleh kembali file tersebut dan memasukkan file tersebut sebagai bagian dari time line berdasarkan *MAC time*. Jika *MAC time* dari file yang terhapus overlap dengan periode waktu seorang penyerang masuk ke sistem, maka perubahan file tersebut dapat dikaitkan dengan orang tersebut. Jika hanya berkaitan dengan *MAC time* suatu file tunggal, maka instruksi *UNIX stat* dapat digunakan. Instruksi *stat* ini menampilkan beberapa jenis informasi tentang file yang mengandung *MAC time*.

```
[root@whammo /var] mac-robber /log >
body.mac
[root@whammo /var] mactime -b
body.mac 9/1/2002 > timelime
Tue Sep 10 2002 15:21:34
63344 ..c -rw-r--r-- 0 0 1418328
log/ksyms.3
63344 mac -rw-r--r-- 0 0 1418346
log/ksyms.0
63344 ..c -rw-r--r-- 0 0 1418334
log/ksyms.1
[root@whammo]# stat
idaho.network.forensics.paper
File: "idaho.network.forensics.paper"
Size: 17180 Blocks: 40 Uid:
(500/jpc) Gid: (500/jpc) Device:
305h/773d
Inode: 669381 Links: 1
Access: (0664/-rw-rw-r--)
Access: Fri Aug 30 08:35:57 2002
Modify: Fri Aug 30 08:35:57 2002
Change: Fri Aug 30 08:39:39 2002
```

Contoh di atas di mana waktu akses dan waktu modifikasi yang sama serta waktu perubahan 4 menit kemudian menunjukkan perubahan kepemilikan atau izin setelah file dibuat. Juga ditunjukkan pemilik *file*, ukuran, perijinan, jumlah blok yang digunakan, nomor inode dan jumlah link ke *file*.

2.5. Tool dan Toolkit Forensik

Tool dan *toolkit* merupakan seperangkat *software*, dan *hardware* pada beberapa kasus, untuk membantu melakukan proses forensik. Sistem operasi yang digunakan sangat mempengaruhi pilihan *toolkit* forensik yang diperlukan. *Tool* forensik yang paling *powerful* adalah **EnCase** (untuk Windows, www.encase.com). Tetapi karena harganya cukup mahal (sekitar \$2,500), sebagai alternatifnya digunakan Linux sebagai platform forensik, di mana tersedia tool gratis, lagipula berbagai sistem file disupport dengan Linux. *Linux forensic tool* memungkinkan melaksanakan analisa forensik pada Windows, Linux, BSD, DOS, dsb.

Sekumpulan tool forensik *berplatform Linux* dikenal dengan *The Coroner's Toolkit (TCT)* dan pelengkapya adalah :

- *TCTUtils*
- *Autopsy Forensic Browser*

2.5.1. The Coroner's Toolkit (TCT)

TCT, dibuat oleh Dan Farmer dan Wietse Venema,, merupakan sekumpulan tool berbasis Linux yang merupakan teknik yang *powerful* untuk mengumpulkan dan menganalisa data forensik. Tujuannya adalah merekonstruksi kejadian yang lalu dan memulihkan/recoveri data yang terhapus. Toolkit ini dapat untuk menganalisa pada sistem *online* maupun *offline*.

TCT terdiri dari sekelompok tool, yakni :

- (1) *grave-robber*: Menangkap berbagai tipe data secara cepat dan membuat *MD5 hash* untuk menjaga integritas data. Informasi yang ditangkap meliputi *MAC time*, file yang terhapus dan isi memori.
- (2) *mac-robber*: mengganti *grave-robber* secara fungsional dengan membangkitkan *MAC time*.
- (3) *pcat, ils, icat, file*: mencatat dan menganalisa proses dan data *inode*.
 - *Pcat* : mengkopi memori proses dari live system
 - *Ils* : menampilkan informasi inode
 - *Icat* : mengkopi file dengan nomor inode
 - *File* : mengklasifikasi file dalam berbagai jenis.
- (4) *Unrm dan Lazarus* : Memulihkan dan menganalisa *unallocated disk blocks* pada sistem file.

- *Unrm* : mengumpulkan informasi pada bagian sistem file yang tidak ditentukan
- *Lazarrus* : menganalisa data kasar pada unrm dan mengklasifikasikan jenis datanya
- *Mactime* : digunakan untuk membuat time line tentang kapan file dimodifikasi, diakses atau dibuat.

2.5.2. TCTUtils

Yaitu sekumpulan *utility* yang menambah fungsi dari *TCT*, terdiri dari :

- (1) *Bcat* : menampilkan blok disk ke *stdout*
- (2) *Blockcalc* : memetakan *image dd* dan hasil *unrm*
- (3) *Fls* : menampilkan file dan direktori yang telah dihapus
- (4) *Find-file* : menentukan file yang mana yang dialokasikan *inode*

2.5.3. Autopsy Forensic Browser

Merupakan *GUI front end* ke *TCT* dan *TCTUtils*, di mana analis forensik bisa melihat-lihat dan menganalisa citra forensik pada file , blok dan inode, dan juga melakukan pencarian dengan *keyword*. Tool ini bisa digunakan sebagai alternatif dari tool forensik berbasis Windows, seperti *EnCase*.

BAB III

PENGELOMPOKAN TINGKAT SIGNIFIKANSI INPUT

Pengelompokan *feature* berdasarkan tingkat signifikansi sangatlah penting bagi proses forensik karena akan meningkatkan ketelitian dan mempercepat proses sehingga secara keseluruhan kinerja forensik juga meningkat.

Input dikelompokkan menjadi input yang penting, kurang penting dan tidak berguna. Input yang tidak berguna dibuang, dan kelompok input terpenting mendapat prioritas untuk dianalisa terlebih dahulu. Dengan demikian akan sangat menghemat waktu analisa karena pada umumnya volume input sangatlah besar.

3.1. Metoda Pengelompokan Berdasarkan Kinerja

Secara umum metoda untuk mengelompokkan input berdasarkan kinerjanya adalah sebagai berikut [1] :

(1) Susun *training set* dan *testing set*.

Untuk setiap *feature* dilakukan hal-hal berikut :

(2) Hapus *feature* dari data (*training* dan *testing*).

(3) Gunakan data yang tersisa untuk melatih *classifier*.

(4) Analisa kinerja *classifier* dengan menggunakan *testing set* sesuai dengan kriteria kinerja yang dipilih.

(5) Kelompokkan *feature* berdasarkan tingkat signifikansi sesuai aturannya.

3.1.1. Metrik Kinerja untuk Pengelompokan Berbasis *Support Vector Machine (SVM)*

Kinerja yang dijadikan metrik/ukuran pengelompokan berbasis SVM yaitu :

➤ Ketepatan (*accuracy*)

➤ *Training time*

➤ *Testing time*

Setiap *feature* akan digolongkan menjadi *feature* yang penting, sekunder atau tidak penting.

Aturan pengelompokan berbasis *SVM* adalah sebagai berikut [1] :

(1) Jika *accuracy* turun dan *training time* naik dan *testing time* turun maka *feature* tergolong penting.

(2) Jika *accuracy* turun dan *training time* naik dan *testing time* naik maka *feature* tergolong penting

(3) Jika *accuracy* turun dan *training time* turun dan *testing time* naik maka *feature* tergolong penting

- (4) Jika *accuracy* tidak berubah dan *training time* naik dan *testing time* naik maka *feature* tergolong penting
- (5) Jika *accuracy* tidak berubah dan *training time* turun dan *testing time* naik maka *feature* tergolong sekunder
- (6) Jika *accuracy* tidak berubah dan *training time* naik dan *testing time* turun maka *feature* tergolong sekunder
- (7) Jika *accuracy* tidak berubah dan *training time* turun dan *testing time* turun maka *feature* tergolong tidak penting
- (8) Jika *accuracy* naik dan *training time* naik dan *testing time* turun maka *feature* tergolong sekunder
- (9) Jika *accuracy* naik dan *training time* turun dan *testing time* naik maka *feature* tergolong sekunder
- (10) Jika *accuracy* naik dan *training time* turun dan *testing time* turun maka *feature* tergolong tidak penting

3.1.2. Metrik Kinerja untuk Pengelompokan Berbasis *Neural Network*

Kinerja yang dijadikan metrik/ukuran pengelompokan berbasis *ANN* yaitu :

- Ketepatan keseluruhan (*overall accuracy*)/ *OA* dari 5 kelas serangan
- *False positif rate* / *FP*
- *False negative rate* / *FN*

Setiap *feature* akan digolongkan menjadi *feature* yang penting, sekunder atau tidak penting.

Aturan pengelompokan berbasis *ANN* adalah sebagai berikut [1] :

- (1) Jika *OA* naik dan *FP* turun dan *FN* turun maka *feature* tergolong tidak penting.
- (2) Jika *OA* naik dan *FP* naik dan *FN* turun maka *feature* tergolong tidak penting.
- (3) Jika *OA* turun dan *FP* naik dan *FN* naik maka *feature* tergolong penting
- (4) Jika *OA* turun dan *FP* turun dan *FN* naik maka *feature* tergolong penting
- (5) Jika *OA* tidak berubah dan *FP* tidak berubah maka *feature* tergolong sekunder.

Metrik kinerja dan aturan pengelompokan tidak dibatasi hanya seperti aturan tersebut diatas, tetapi dapat berubah sesuai kompleksitas dan sifat masalah.

3.2. Metoda Pengelompokan Berdasarkan *Support Vector Decision Function (SVDF)*

Dengan menggunakan *Support Vector Decision Function/ SVDF*, yang menyimpan informasi tentang *feature* dan kontribusinya terhadap klasifikasi, input dapat digolongkan berdasarkan tingkat signifikansinya. Persamaan fungsi tersebut sbb :

$$\mathbf{F}(\mathbf{X}) = \sum \mathbf{W}_i \mathbf{X}_i + \mathbf{b}$$

\mathbf{X} termasuk kelas positif jika $\mathbf{F}(\mathbf{X})$ positif

\mathbf{X} termasuk kelas negatif jika $\mathbf{F}(\mathbf{X})$ negatif

\mathbf{W}_i : ukuran kekuatan klasifikasi

Jika \mathbf{W}_i bernilai positif besar maka feature ke- i merupakan faktor kunci untuk kelas positif

Jika \mathbf{W}_i bernilai negatif besar maka feature ke- i merupakan faktor kunci untuk kelas negatif

Jika \mathbf{W}_i bernilai mendekati nol baik dari sisi positif maupun negatif maka feature ke- i tidak cukup berarti bagi klasifikasi

Berdasarkan fungsi tersebut, input dapat dikelompokkan menggunakan prosedur berikut :

- (1) Data asli digunakan untuk melatih classifier
- (2) *SVDF* digunakan untuk mengelompokkan input berdasarkan tingkat signifikansi dengan cara :
- (3) Hitung pemberatan */weights* dari *SVDF*
- (4) Golongkan tingkat signifikansi input berdasarkan nilai absolut pemberatannya

3.3. Perbandingan Ekstraksi *Significant Feature* Menggunakan *SVM* vs *ANN*

Berdasarkan referensi [1] kedua metoda pengelompokan input, yakni *SVM* dan *ANN*, telah diujikan pada data deteksi intrusi DARPA 1998 yang berasal dari *MIT 's Lincoln Lab* yang terdiri dari 5 kelas data. Data diklasifikasi menjadi 5 kelas yang terdiri dari satu kelas data pada keadaan jaringan normal dan 4 kelas serangan, yakni :

- (1) Normal
- (2) *Probing* : suatu kelas serangan di mana penyerang mengamati suatu jaringan komputer untuk mengumpulkan informasi atau menemukan kelemahan jaringan/lubang keamanan jaringan, dengan menggunakan peta jaringan dan layanan yang tersedia pada jaringan, misal *Ipsweep*, *Mscan*, *Nmap*, *Saint*, *Satan*.
- (3) *Denial of Service Attacks* : suatu kelas serangan dimana penyerang menjadikan server jaringan sibuk melayani permintaan yang dibuatnya sehingga menolak melayani permintaan user lainnya. Misal *Apache2*, *Back*, *Land*, *Mail bomb*, *SYN Flood*, *Ping of death*, *Process table*, *Smurf*, *Syslogd*, *Teardrop*, *Udpstorm*.
- (4) *User to Root Attacks /U2Su* : suatu kelas serangan dimana penyerang mula-mula mengakses jaringan dengan menggunakan account yang sah dan mengeksploitasi kelemahan yang ada dan kemudian mengakses jaringan secara lebih dalam. Misalnya *Eject*, *Ffbconfig*, *Fdformat*, *Loadmodule*, *Perl*, *Ps*, *Xterm*.

(5) *Remote to Local Attacks / R2L* : suatu kelas serangan di mana penyerang yang tidak mempunyai account yang sah mengeksploitasi kelemahan jaringan untuk mengakses jaringan dan mengirim paket ke suatu komputer melalui jaringan. Contohnya adalah *Dictionary, Ftp_write, Guest, Imap, Named, Phf, Sendmail, Xlock, Xsnoop*.

Unjuk kerja dari hasil pengelompokan berdasarkan tingkat signifikansi menggunakan kedua metoda pengelompokan input, yakni *SVM* dan *ANN* yang telah diujikan pada data deteksi intrusi DARPA 1998 yang berasal dari *MIT 's Lincoln Lab* dapat dilihat pada tabel 1 sampai 8.[1]

Tabel 1. Kinerja dari *SVM* menggunakan seluruh data (41 features)

Class	Training Time (sec)	Testing Time (sec)	Accuracy (%)
Normal	7.66	1.26	99.55
Probe	49.13	2.10	99.70
DOS	22.87	1.92	99.25
U2Su	3.38	1.05	99.87
R2L	11.54	1.02	99.78

Tabel 2. Kinerja dari *SVM* menggunakan hanya *feature* penting

Class	No of Features	Training Time (sec)	Testing Time (sec)	Accuracy (%)
Normal	25	9.36	1.07	99.59
Probe	7	37.71	1.87	99.38
DOS	19	22.79	1.84	99.22
U2Su	8	2.56	0.85	99.87
R2L	6	8.76	0.73	99.78

Tabel3: Kinerja dari *SVM* menggunakan perpaduan *feature* penting (30)

Class	Training Time (sec)	Testing Time (sec)	Accuracy (%)
Normal	7.67	1.02	99.51%
Probe	44.38	2.07	99.67%
DOS	18.64	1.41	99.22%
U2Su	3.23	0.98	99.87%
R2L	9.81	1.01	99.78%

Tabel 4. Kinerja dari *SVM* menggunakan *feature* penting dan sekunder

Class	No of Features	Training Time (sec)	Testing Time (sec)	Accuracy (%)
Normal	39	8.15	1.22	99.59
Probe	32	47.56	2.09	99.65
DOS	32	19.72	2.11	99.25
U2Su	25	2.72	0.92	99.87
R2L	37	8.25	1.25	99.80

Tabel 5. Kinerja dari *SVM* menggunakan hanya *feature* penting dengan metoda *SVDF*

Class	No of Features	Training Time (sec)	Testing Time (sec)	Accuracy (%)
Normal	20	4.58	0.78	99.55
Probe	11	40.56	1.20	99.36
DOS	11	18.93	1.00	99.16
U2Su	10	1.46	0.70	99.87
R2L	6	6.79	0.72	99.72

Tabel 6. Kinerja dari *SVM* menggunakan perpaduan *feature* penting (23) dengan metoda *SVDF*

Class	Training time	Testing time	Accuracy (%)
Normal	4.85	0.82	99.55
Probe	36.23	1.40	99.71
DOS	7.77	1.32	99.20
U2Su	1.72	0.75	99.87
R2L	5.91	0.88	99.78

Tabel 7. Kinerja dari *SVM* menggunakan *feature* penting dan sekunder dengan metoda *SVDF*

Class	No of Features	Training Time (sec)	Testing Time (sec)	Accuracy (%)
Normal	34	4.61	0.97	99.55
Probe	21	39.69	1.45	99.56
DOS	19	73.55	1.50	99.56
U2Su	23	1.73	0.79	99.87
R2L	20	5.94	0.91	99.78

Tabel 8. Kinerja dari *ANN* menggunakan seluruh data (41 *features*) dan 34 *feature* penting

Number of features	Accuracy	False positive rate	False negative rate	Number of epochs
41	87.07	6.66	6.27	412
34	81.57	18.19	0.25	27

Dari tabel-tabel di atas terlihat bahwa dengan menggunakan *SVM* (dengan metrik kinerja ataupun *SVDF*) *accuracy*-nya di atas 99 %, baik menggunakan seluruh data ataupun hanya data penting atau data penting dan sekunder. Sedangkan dengan menggunakan *ANN* *accuracy*-nya di bawah 90 %.

Disamping *accuracy* yang lebih tinggi, *SVM* mempunyai skalabilitas yang lebih baik dibandingkan dengan *ANN*, yakni *SVM* dapat melatih sejumlah besar pola, sedangkan *ANN* membutuhkan waktu yang lebih lama bahkan terkadang gagal ketika jumlah pola cukup besar[1]. Juga *SVM* lebih cepat (training time dan testing time) daripada *ANN* [1].

BAB IV

KESIMPULAN

Network Forensic diperlukan untuk mengantisipasi kian maraknya *cyber criminal*, namun masih terdapat banyak kelemahan dalam teknik maupun toolnya, yang sering dimanfaatkan oleh para penjahat cyber yang kian pintar. Dengan demikian diperlukan riset untuk mengembangkan teknik dan tool forensik jaringan, di mana dalam hal ini berlomba dengan penjahat dunia maya tersebut. Kekalahan dalam perlombaan ini akan menimbulkan kerugian besar dan ancaman yang serius karena saat ini dan mendatang jaringan komputer merupakan inti dari hampir semua kegiatan.

Salah satu teknik dan tool yang perlu dikembangkan adalah otomatisasi ekstraksi ciri signifikan pada analisa forensik jaringan untuk meningkatkan *accuracy* dan mempercepat prosesnya.

Mesin pembelajaran *SVM* sebagai salah satu teknik intelligensia buatan dapat digunakan untuk ekstraksi ciri signifikan tersebut dengan akurasi yang tinggi (di atas 99%) untuk seluruh kelas (5 kelas) dan mempunyai kecepatan yang tinggi serta skalabilitas yang lebih baik dibanding teknik intelligensia lainnya yaitu *ANN* jika digunakan untuk analisa forensik jaringan offline [1].

DAFTAR PUSTAKA

- [1] Mukkamala, Srinivas & Andrew H. Sung .”Identifying Significant Features for Network Forensic Analysis Using Artificial Intelligent Techniques” . *International Journal of Digital Evidence, Volume 1, Issue 4 Winter 2003*. www.ijde.org/docs/02_winter_art3.pdf
- [2] Dittrich, D. (2001). “Basic Steps in Forensic Analysis of Unix Systems”.
<http://staff.washington.edu/dittrich/misc/forensics>
- [3] Farmer, D., & Venema, W. (1998). “The Coroner’s Toolkit”. <http://www.porcupine.org>.
- [4] Vatis, Michael A. (2002). “Law Enforcement Tool and Technologies for Investigating Cyber Attacks”. www.ists.dartmouth.edu
- [5] The O'Reilly Network.”Network Forensics: Tapping the Internet”.
<http://www.oreillynet.com/pub/a/network/2002/04/26/nettap.html>
- [6] La Bancz, Melisa (2003). “Making It Big : Large Scale Network Forensics”.
www.LinuxSecurity.com
- [7] Yasinsac, Alec & Yanet Manzano. “Honeytraps, A Network Forensic Tool”.
<http://www.cs.fsu.edu/~yasinsac/Papers/YM02.pdf>
- [8] “Cyber Cop Sting”, <http://www.nai.com>