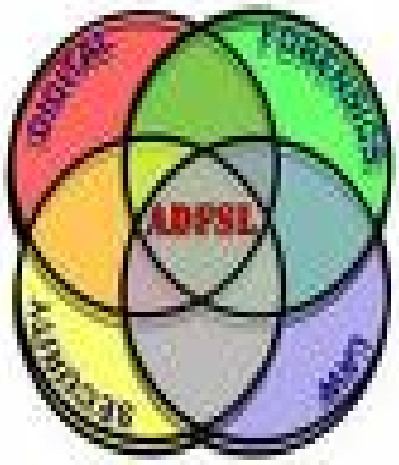



**Achmad Benny Mutiara**

# *Panduan Praktis dan Terpadu Forensik Teknologi Informasi*

Penerbit Universitas Gunadarma

*Korespondensi:*

Dosen Universitas Gunadarma
Alamat: Jl. Walet No.24, Bogor 16161
Telp: 0251 321796
Hp: 08159116055
e-mail: amutiara@staff.gunadarma.ac.id ,
amutiara@gmail.com
Blog: http://abmutiara.info/? ,

Kontributor: April, mahasiswa bimbingan penulisan ilmiah, Jurusan Teknik Informatika Universitas Gunadarma

# Kata Pengantar

Pada abad ke 21 ini perkembangan teknologi sudah semakin pesat. Terutama teknologi informasi yang tiap harinya mengalami kemajuan. Penggunaan teknologi informasi juga sudah merambah ke setiap sendi kehidupan manusia.

Contohnya, setiap organisasi membutuhkan banyak data untuk diolah menjadi sebuah informasi yang berguna bagi organisasi tersebut. Data seperti ini biasa disimpan atau di transfer oleh sistem komputer, *personal digital assistants* (PDA), *networking equipment* dan sumber – sumber data lainnya. Untuk itu penting adanya analisa data. Karena analisa data dapat digunakan untuk berbagai tujuan, seperti merekonstruksi kejadian keamanan komputer, *trouble shooting* permasalahan operasional dan pemulihan dari kerusakan sistem yang terjadi secara mendadak.

Analisa data terdiri dari analisa data komputer dan analisa data jaringan. Analisa data komputer berhubungan dengan data pada media penyimpanan suatu komputer, sedangkan analisa data jaringan berhubungan dengan data yang melintas pada suatu jaringan. Jadi, jika kedua jenis analisa ini dikombinasikan maka dapat menangani dan memberikan dukungan operasional terhadap suatu masalah.

Untuk melakukan analisa ini, ada proses – proses yang harus dilakukan, diantaranya *acquisition, examination, utilization dan review*. Biasanya dalam melakukan proses - proses tersebut terdapat kesulitan, untuk itu penulis membuat *guide lines* proses – proses tersebut. Hal ini diupayakan untuk memudahkan penyelenggaraan proses analisa data tersebut, serta untuk memberikan informasi atas penggunaan proses dengan empat kategori sumber data utama, diantaranya file, sistem operasi, lalu lintas jaringan dan aplikasi.

Tujuan dari penulisan buku ini ialah memberikan suatu standar *guide lines,* supaya kita dapat menentukan *tools* yang seperti apa yang dapat digunakan dalam menghadapi suatu masalah berdasarkan sumber data yang kita gunakan, karena hal ini merupakan bagian dari proses analisa data.

Ucapan terima kasih penulis sampaikan pada Pimpinan Universitas Gunadarma, Mahasiswa Teknik Informatika, serta isteri dan anak-anakku tercinta, karena atas dorongan dan semangat dari mereka buku ini tertulis.

Buku ini masih jauh dari sempurna. Kritik dan saran yang membangun dapat para pembaca sampaikan ke penulis.

Semoga buku ini bermanfaat bagi para ahli di bidang teknologi informasi yang berminat pada bidang spesifik forensik digital. Amin.

Depok, 2007

Penulis
ABM

# DAFTAR ISI

# Ringkasan Eksekutif

Istilah data mengacu pada bagian informasi digital berbeda yang telah diformat secara spesifik. Organisasi – organisasi menambahkan sejumlah data dari berbagai sumber, sebagai contoh mungkin saja dari data yang disimpan ataupun yang dikirim melalui sistem komputer standar, peralatan *networking*, alat bantu komputer, PDA, alat elektronik yang digunakan oleh konsumen serta sumber data lainnya. Analisa data dapat digunakan untuk berbagai macam tujuan, seperti merekontruksi peristiwa keamanan komputer, memecahkan masalah operasional dan mengembalikan lagi data dari kerugian yang diakibatkan kerusakan pada sistem. Biasanya setiap organisasi membutuhkan kemampuan untuk mendayagunakan komputer dan menganalisa data jaringan.Untuk kesepakatan, panduan ini menyediakan informasi yang lengkap dalam membentuk kemampuan menganalisis data, termasuk membangun peraturan - peraturan dan prosedur – prosedur.

Menurut kebiasaannya, analisa data komputer dihubungkan dengan data pada media penyimpanan komputer, sedangkan untuk analisa data jaringan dihubungkan dengan data yang melintas pada suatu jaringan. Sebagai alat dan teknik analisa yang sering digunakan, kedua displin ini sudah terjalin. Kombinasi antara kemampuan analisis data komputer dan jaringan sangat penting untuk menangani suatu kejadian dan sebagai pendukung operasional. Untuk kedua analisis data yaitu analisis data komputer dan jaringan, maka proses analisa terdiri atas tahap – tahap berikut :

1. *Acquisition* (didapatnya) : memperoleh data dari sumber yang mungkin untuk data yang relevan, serta memeriksakan prosedur untuk integritas data dari sumber data.
2. *Examination* (pengujian) : penggunaan metode otomatis untuk menyelidiki data yang diperoleh .
3. *Utilization* (pemanfaatan) : laporan dari hasil pengujian, yang mana meliputi penggunaan tindakan dalam pengujian dan saran untuk peningkatan.
4. *Review* (tinjauan ulang) : melakukan tinjauan ulang untuk proses dan praktek dalam konteks tugas yang sekarang untuk mengidentifikasi kekurangbijakan, kesalahan prosedur dan permasalahan lain yang perlu untuk ditinjau ulang. Pelajaran untuk mempelajari pada sepanjang tahap tinjauan ulang harus disatukan kedalam usaha analisa data berikutnya.

Panduan ini menyediakan saran yang bersifat umum untuk menyelenggarakan proses analisa data. Selain itu juga menyediakan informasi yang detil untuk memakai proses pada empat kategori sumber data utama : file – file, sistem operasi, *network traffic* dan berbagai aplikasi lainnya. Panduan ini fokus pada menjelaskan karakteristik dan komponen – komponen dasar dari sumber data pada setiap kategori. Panduan ini juga menyediakan saran untuk bagaimana caranya sumber data yang multiple dapat digunakan bersama – sama dalam mendapatkan suatu pemahaman yang lebih baik dari suatu kejadian.

Proses dan teknik analisa data yang diperkenalkan di dalam pemandu ini didasarkan pada prinsipnya forensik digital. Ilmu pengetahuan forensik biasanya digambarkan sebagai aplikasi ilmu pengetahuan terhadap hukum. Forensik digital mempunyai banyak definisi karena juga dikenal sebagai jaringan dan komputer forensik. Biasanya, forensik digital dianggap sebagai aplikasi ilmu pengetahuan untuk mengidentifikasi, mengoleksi, menganalisa dan menguji data - data digital ketika pemeliharaan integritasnya dari informasi dan pemeliharaan serangkaian penjagaan yang tegas untuk data – data tersebut. Analisa data jaringan dan komputer serupa dengan forensik digital dan menggunakan banyak dari *tools* dan teknik yang sama, tetapi analisa data tidak perlu meliputi semua tindakan yang penting untuk memelihara integritas dari semua informasi yang dikumpulkan, atau pun melakukan hal lain meliputi pemeliharaan suatu rantai penjagaan atau tindakan pemeliharaan bukti lain. **Oleh karena itu, penerbitan ini tidak boleh digunakan sebagai suatu panduan untuk pelaksanaan suatu penyelidikan forensik digital, menerangkan dasar - dasar yang sah menurut hukum atau menggunakannya sebagai dasar untuk melakukan aktifitas kriminal dalam penyelidikan.**

Menerapkan rekomendasi berikut dianggap perlu untuk memudahkan aktifitas analisa data jaringan dan komputer menjadi lebih efektif dan efisien untuk para agen dan departemen pemerintah pusat.

**Organisasi perlu memastikan bahwa kebijakan mereka berisi statemen jelas yang mengarah ke semua pertimbangan analisa data utama, seperti melakukan *monitoring* (pengawasan) serta melaksanakan tinjauan ulang prosedur dan kebijakan analisa data yang reguler.**

Pada tingkatan yang lebih tinggi, kebijakan perlu mengijinkan personil diberi hak untuk memonitor jaringan dan sistem, juga melaksanakan penyelidikan untuk pertimbangan yang sah di bawah keadaan yang sesuai. Organisasi juga boleh mempunyai suatu kebijakan

analisa data terpisah untuk penanganan suatu peristiwa dan hal yang lainnya dengan peran analisa data yang menyediakan lebih terperinci aturan untuk peristiwa yang sesuai. Kebijakan analisa data perlu dengan jelas menggambarkan tanggung-jawab dan peran dari semua orang dan organisasi eksternal dalam menyelenggarakan atau membantu aktifitas analisa data pada organisasi. Kebijakan ini juga perlu dengan jelas menandai siapa saja yang perlu dihubungi dalam regu internal dan organisasi eksternal di bawah keadaan yang berbeda.

**Organisasi perlu menciptakan dan memelihara prosedur untuk menyelenggarakan tugas analisa data, berdasarkan kebijakan pada organisasi dan peserta yang melakukan aktifitas analisa data, seperti halnya semua ketentuan hukum dan peraturan yang bisa diterapkan.**

Prosedur perlu memusatkan pada metodologi umum untuk menyelidiki peristiwa yang menggunakan teknik analisa data, karena tidak mungkin untuk mengembangkan prosedur secara menyeluruh yang dikhususkan ke tiap-tiap situasi yang mungkin. Bagaimanapun, organisasi perlu mempertimbangkan pengembangan tahapan – tahapan prosedur untuk melakukan tugas yang rutin. Prosedur harus bersifat konsisten, efektif dan merupakan tindakan analisa data yang akurat. Prosedur tersebut harus ditinjau secara periodik, seperti halnya terdapat perubahan penting yang dibuat oleh tim kebijakan dan prosedur.

**Organisasi perlu memastikan bahwa kebijakan dan prosedur mereka mendukung suatu alasan dan layak menggunakan *tools* untuk menganalisa data.**

Prosedur dan kebijakan organisasi harus dengan jelas menerangkan tentang tindakan analisa data yang perlu dan tidak perlu dilakukan diberbagai keadaan, seperti halnya menguraikan surat pengantar yang perlu untuk informasi yang sensitif yang boleh jadi direkam oleh alat analisa, seperti kata sandi, data pribadi (contoh angka-angka jaminan sosial) dan isi dari e-mail. Penasehat hukum perlu secara hati-hati meninjau ulang semua kebijakan analisa data dan prosedur tingkat tinggi.

**Organisasi perlu memastikan bahwa para profesional IT mereka disiapkan untuk mengambil bagian di dalam aktifitas analisa data.**

Para profesional IT setiap suatu organisasi, terutama orang yang menangani kejadian dan yang pertama kali merespon, perlu memahami tanggung-jawab dan peran mereka untuk analisa data, menerima pendidikan dan pelatihan tentang prosedur dan kebijakan analisa data yang terkait, dan disiapkan untuk bekerja sama dan membantu yang lain manakala teknologi mereka bertanggung jawab untuk menjadi bagian dari dari suatu peristiwa atau kejadian

lainnya. Para profesional IT perlu juga berkonsultasi dengan penasehat yang sah tentang undang-undang secara lebih dekat, kedua hal ini ada di dalam persiapan umum untuk aktifitas analisa data, seperti menentukan tindakan para profesional IT yang tidak dan perlu dilaksanakan dan juga jika dibutuhkan suatu dasar untuk mendiskusikan situasi analisa data yang spesifik. Juga, manajemen harus bertanggung jawab untuk mendukung kemampuan analisa data, menyetujui dan meninjau ulang kebijakan analisa data dan menyetujui tindakan analisa data tertentu, seperti mengambil *mission-critical* sistem off-line.

# 1 Pengenalan

## 1.1 Otoritas

Institut Nasional Teknologi dan Standard ( NIST) mengembangkan dokumen ini lebih jauh tentang undang-undang tanggung-jawab di bawah Federal Information Security Management Act ( FISMA) 2002, Hukum publik 107-347.

NIST bertanggung jawab untuk mengembangkan petunjuk dan standard, termasuk kebutuhan minimum, untuk menyediakan keamanan informasi yang cukup untuk semua asset dan operasi agen, tetapi petunjuk dan standard seperti ini tidak dapat diberlakukan pada sistem keamanan nasional. Petunjuk ini konsisten dengan kebutuhan dari Office of Management and Budget ( OMB) Lingkar A-130, Bagian 8b(3), " Pengamanan Sistem informasi Agen," seperti yang dianalisa di dalam A-130, Catatan tambahan IV: Analisa Bagian Kunci. Informasi bersifat tambahan disediakan dalam A-130, Catatan tambahan III.

Petunjuk ini telah disiapkan untuk penggunaan oleh para agen Pemerintah pusat. Petunjuk ini mungkin saja digunakan oleh organisasi non pemerintah yang termasuk dalam golongan sukarela dan tidak ditujukan kepada hak cipta, meskipun sumbernya diinginkan.

Tidak ada apapun di dalam dokumen ini yang harus diambil untuk membantah petunjuk dan standard ini dapat membuat para agen Pemerintah pusat memiliki hak ataupun ditekan oleh Sekretaris Perdagangan yang berada di bawah undang-undang otoritas, walaupun harus begitu petunjuk ini diinterpretasikan dengan mengubah atau menggantikan otoritas yang ada dari Sekretaris Perdagangan, Direktur (menyangkut) OMB, atau pejabat Pemerintah pusat lain.

## 1.2 Lingkup dan Tujuan

Penerbitan ini ditujukan untuk mencari organisasi supaya dapat dibantu di dalam penyelidikan pada peristiwa keamanan komputer dan *troubleshooting* permasalahan operasional teknologi informasi ( IT) dengan menyediakan bimbingan praktis pada penelitian data dari komputer dan jaringan. Yang secara rinci, dokumen meliputi suatu uraian dari proses untuk menyelenggarakan analisa data yang efektif, dan juga menyediakan saran

mengenai perbedaan sumber data, termasuk didalamnya adalah file, sistem operasi, *network traffic*, dan aplikasi.

Penerbitan ini tidak boleh digunakan sebagai suatu panduan untuk pelaksanaan suatu penyelidikan forensik digital, menerangkan dasar - dasar yang sah menurut hukum atau menggunakannya sebagai dasar dalam melakukan aktifitas kriminal dalam penyelidikan. Tujuannya adalah menginformasikan kepada pembaca tentang berbagai teknologi dan potensi apa saja yang dapat digunakan oleh mereka manakala melakukan aktifitas *troubleshooting* atau menanggapi suatu kejadian. Pembaca disarankan untuk hanya menerapkan langsung apa saja yang direkomendasikan setelah konsultasi dengan manajemen dan penasehat yang sah tentang undang-undang dalam pemenuhan peraturan dan hukum (yaitu lokal, status, pemerintah pusat dan internasional) yang menyinggung kepada situasi yang mereka hadapi.

## 1.3 Audience

Dokumen ini telah diciptakan untuk tim yang menanggapi kejadian; sistem, jaringan dan pengurus keamanan; dan para manajer program keamanan komputer yang bertanggung jawab untuk memperoleh dan menguji data untuk menanggapi suatu peristiwa atau bertujuan untuk *troubleshooting*. Praktek yang direkomendasikan di dalam panduan ini dirancang untuk menyoroti kunci dari prinsip yang dihubungkan dengan analisa data jaringan dan komputer.

Oleh karena perubahan alami yang secara konstan pada alat analisa dan sumber data jaringan dan komputer, pembaca diharapkan untuk mengambil keuntungan dari sumber daya yang lain, mencakup yang disebutkan dalam panduan ini, untuk informasi detail dan lebih jauhnya akan diperkenalkan di dalam panduan ini.

## 1.4 Struktur Dokumen

Dokumen ini diorganisir ke dalam tujuh bagian utama berikut :

1. Bagian 2 mendiskusikan kebutuhan analisa data jaringan dan komputer dan juga menyediakan bimbingan pada pendirian akan kemampuan analisa data untuk suatu organisasi.
2. Bagian 3 menjelaskan langkah-langkah dasar yang dilibatkan di dalam menyelenggarakan proses analisa data jaringan dan komputer: didapatnya data, pengujian, pemanfaatan dan tinjauan ulang.
3. Bagian 4 sampai 7 menyediakan detil tentang memperoleh dan menguji data dari berbagai sumber data, berdasar pada kerangka di dalam Bagian 3. Kategori sumber

data yang dibahas berturut-turut di dalam Bagian 4 sampai 7 adalah file data, sistem operasi, *network traffic* dan aplikasi.

4. Bagian 8 diberikan banyak contoh studi kasus yang menggambarkan bagaimana analisa dapat menghubungkan peristiwa antar beberapa sumber data.

Dokumen juga berisi beberapa catatan-catatan tambahan dengan pendukung isi dari panduan ini, diantaranya:

5. Catatan tambahan yang diberikan berupa rekomendasi yang utama dan dibuat sepanjang dokumen ini.
6. Catatan tambahan B memberikan skenario dimana teknik analisa data mungkin dapat bermanfaat dan pertanyaan untuk pembaca pada satu rangkaian pertanyaan mengenai masalahnya masing-masing.
7. Catatan-catatan tambahan C dan D berturut-turut berisi suatu daftar kata dan singkatan daftar.

# Membangun dan Mengorganisir suatu Kemampuan Analisa Data

# 2

Istilah data mengacu pada bagian - bagian yang berbeda dari informasi digital yang telah diformat secara spesifik. Perluasan dari komputer[1] untuk penggunaan pribadi dan profesional dan hal mudah dalam menembus jaringan memiliki bahan yang dibutuhkan supaya *tools* dapat meneliti data dari banyak sumber dimana jumlah datanya selalu meningkat. Sebagai contoh, data mungkin disimpan atau ditransfer oleh sistem komputer standard (desktop, *laptops*, server, dsb), peralatan *networking* (*firewalls, routers*, dsb), *personal digital assistant* ( PDA), CD, DVD, *removables hard drives, backup tape, flash memory, thumbs drives dan jump drives*. Banyak barang elektronika yang dipakai konsumen dapat digunakan untuk menyimpan data; contohnya telepon selular, *video game console, digital audio player* dan perekam video digital. Hal ini meningkatkan variasi dari sumber data yang telah membantu untuk memacu perbaikan dan pengembangan alat dan teknik analisis data. Hal ini telah pula disebabkan oleh realisasi dari teknik dan alat seperti itu yang dapat digunakan untuk banyak tujuan, seperti merekonstruksi peristiwa keamanan komputer, *troubleshooting operational problems* dan memperbaiki dari kerusakan sistem yang terjadi secara kebetulan.

Bagian ini mendiskusikan beberapa aspek tentang pengaturan suatu kemampuan analisa data untuk suatu organisasi. Hal tersebut dimulai dengan cara memperlihatkan potensi variasi yang luas yang digunakan untuk analisa data dan kemudian memberi suatu ikhtisar tingkat tinggi yang menyangkut proses analisa data. Hal berikutnya adalah bagian yang mendiskusikan bagaimana jasa analisa data secara khusus disediakan dan menyediakan bimbingan dalam hal membangun dan memelihara ketrampilan yang diperlukan dalam melaksanakan tugas untuk menganalisa data. Bagian ini juga menjelaskan kebutuhan yang meliputi berbagai regu dalam seluruh organisasi, seperti penasehat hukum dan staf keamanan fisik dalam beberapa aktifitas analisa data. Bagian akhirnya adalah dengan mendiskusikan bagaimana prosedur dan kebijakan perlu meliputi analisa data, seperti penjelasan tanggung-

jawab dan peran, menyediakan bimbingan atas pemakaian alat dan teknik analisis data yang sesuai dan bergabung dengan analisa data ke dalam daur hidup sistem informasi.

Proses dan teknik analisa data yang diperkenalkan di dalam pemandu ini didasarkan pada prinsipnya forensik digital. Ilmu pengetahuan forensik biasanya digambarkan sebagai aplikasi ilmu pengetahuan kepada hukum tersebut. Forensik digital yang mempunyai banyak definisi juga dikenal sebagai jaringan dan komputer forensik. Biasanya, forensik digital dianggap sebagai aplikasi ilmu pengetahuan untuk mengidentifikasi, mengumpulkan, menganalisa dan menguji bukti digital ketika pemeliharaan integritasnya dari informasi dan pemeliharaan rantai penjagaan yang tegas untuk data – data tersebut. Analisa data jaringan dan komputer hampir sama dengan forensik digital dan juga menggunakan banyak dari *tools* dan teknik yang sama, tetapi analisa data tidak perlu meliputi semua tindakan yang penting bagi pemeliharaan integritas dari semua informasi yang dimiliki atau tidak perlu meliputi serangkaian pemeliharaan atau tindakan keras pemeliharaan bukti. Maka, penerbitan ini tidak boleh digunakan sebagai suatu petunjuk untuk melaksanakan suatu penyelidikan forensik digital, menerangkan seperti nasehat hukum atau menggunakannya sebagai dasar untuk menyelenggarakan kegiatan kriminal dalam penyelidikan.

## 2.1 Kebutuhan akan Analisa Data

Teknik dan analisa data jaringan dan komputer digunakan untuk tujuan seperti berikut:

1. Memecahkan masalah operasional. Banyak teknik dan alat analisa data yang dapat digunakan untuk memecahkan masalah persoalan – persoalan operasional, seperti menemukan fisik dan penempatan sebetulnya dari suatu *host* dengan suatu bentuk konfigurasi jaringan yang salah, memecahkan suatu masalah fungsional dengan suatu aplikasi dan merekam serta meninjau ulang sistem operasi dan aplikasi *configuration settings* yang sedang digunakan untuk *host.*
2. *Log Monitoring*. Berbagai teknik dan *tools* dapat membantu dengan cara memonitoring *log*, seperti analisa tentang *log entries* dan menghubungkan *log entries* ke berbagai sistem yang berseberangan. Hal ini dapat membantu terhadap penanganan suatu peristiwa, mengidentifikasi pelanggaran kebijakan, *auditing*, dan usaha lain.
3. *Data Recovery*. Ada lusinan *tool* yang dapat memulihkan data yang hilang dari sistem. Hal ini meliputi data yang secara kebetulan atau dengan sengaja dihapus, *overwritten*, atau

sebaliknya dimodifikasi. Sejumlah data yang dapat dipulihkan bervariasi berdasarkan pada suatu kasus demi kasus dasar.

4. Memperoleh data. Beberapa organisasi menggunakan *tools* untuk memperoleh data dari hosts yang sedang ditarik dan diatur kembali atau tidak digunakan lagi. Sebagai contoh, manakala seorang *user* meninggalkan suatu organisasi, data dari *user's workstation* dapat diperoleh dan disimpan jika data diperlukan di masa datang. *Media Workstation's* kemudian dapat membersihkan kembali data tersebut untuk kemudian memindahkan semua data asli milik user.

Dengan mengabaikan situasi, proses analisa data terdiri atas tahap berikut :

1. *Acquisition* (perolehan*)*. Tahap yang pertama adalah memperoleh data dari sumber yang mungkin untuk mendapatkan data relevan, diikuti dengan pemeriksaan prosedur lingkungan dari integritas data. Sebagai contoh, *acquisition* pada umumnya dilakukan secara tepat waktu oleh karena ada kemungkinan kehilangan data atau gagal seperti pada koneksi jaringan sekarang.
2. *Examination* (pengujian). Pengujian melibatkan penggunaan metode yang otomatis menyaring dan menggali sejumlah besar data yang diperoleh serta mengidentifikasi fakta dari suatu data tertentu.
3. *Utilization* (pemanfaatan). Tahap selanjutnya adalah melaporkan hasil dari pengujian, yang mana dapat meliputi tindakan yang digunakan di dalam rekomendasi dan pengujian yang ditujukan untuk peningkatan. Formalitas dari langkah pemanfaatan bervariasi sangat tergantung pada situasi tersebut.
4. *Review* (tinjauan ulang). Menyelenggarakan proses tinjauan ulang dan mempraktekkannya dalam konteks tugas yang sekarang dapat mengidentifikasi kekurangbijakan, kesalahan prosedur dan permasalahan lain yang perlu sesuai dengan tujuan. Pelajaran yang mempelajari hal sepanjang tahap tinjauan ulang harus disatukan ke dalam masa depan usaha analisa data.

Bagian 3 menguraikan proses analisa data secara mendalam, sementara itu bagian 4 sampai 7 menyediakan informasi tambahan pada saat proses memperoleh dan menguji jenis data jaringan dan komputer yang berbeda .

## 2.2 Staffing

Pada kenyataannya setiap organisasi harus mempunyai beberapa kemampuan untuk melaksanakan analisa data jaringan dan komputer. Walaupun tingkat dari kebutuhan ini bervariasi, para pengguna utama teknik dan *tools* analisa data di dalam suatu organisasi pada umumnya dapat dibagi menjadi dua kelompok:

1. Para profesional IT. Kelompok ini meliputi pendukung teknis staff dan sistem, jaringan dan *security administrators*. Mereka masing-masing menggunakan sejumlah kecil *tools* dan teknik analisa data yang dikhususkan untuk bagian dari keahlian mereka selama pekerjaan rutin mereka ( seperti *monitoring*, *troubleshooting*, pemulihan data).
2. *Incident Handler*. Mereka bereaksi terhadap berbagai peristiwa keamanan komputer, seperti akses data yang tidak sah, pemakaian sistem yang tidak sesuai, pengaruh kode yang tidak benar dan penolakan atas *service attacks*. Peristiwa penanganan ini secara khusus menggunakan suatu *tools* dan teknik analisa data yang luas selama penyelidikan mereka.

Banyak organisasi bersandar pada suatu kombinasi dari staff mereka sendiri dan bagian - bagian eksternal untuk melaksanakan tugas analisa data. Sebagai contoh, beberapa organisasi melaksanakan tugas analisa data standar sendiri dan menggunakan pihak luar hanya ketika bantuan yang khusus diperlukan. Bahkan organisasi yang ingin melaksanakan semua tugas analisa sendiri pada umumnya paling membutuhkan sumber dari luar, seperti pengiriman media yang telah rusak ke pemulihan data suatu perusahaan supaya dapat direkonstruksi. Tugas seperti itu secara khusus memerlukan penggunaan perangkat lunak, peralatan, fasilitas, dan keahlian teknis yang khusus yang kebanyakan organisasi tidak bisa membenarkan biaya yang tinggi untuk memperoleh dan memeliharanya.

Saat memutuskan bagian – bagian eksternal atau internal mana yang perlu menangani masing-masing aspek dari analisa data, organisasi perlu memikirkan faktor – faktor dibawah ini:

1. Biaya. Ada banyak biaya-biaya yang berpotensi untuk dilibatkan dengan analisa data. Perangkat lunak, perangkat keras dan peralatan yang digunakan untuk memperoleh dan menguji data dapat menimbulkan biaya yang sangat berarti ( misalnya harga pembelian, memperbaharui dan meningkatkan mutu perangkat lunak, pemeliharaan). Biaya-biaya penting lainnya mencakup biaya tenaga kerja dan pelatihan karyawan. Secara umum,

tindakan analisa data yang jarang diperlukan boleh jadi biayanya lebih hemat jika dilakukan oleh pihak luar, sedangkan tindakan yang sering diperlukan boleh jadi biayanya lebih hemat jika dilakukan oleh pihak dalam.

2. *Response Time*. Orang yang ditempatkan pada on site akan mampu memulai aktifitas analisa data dengan cepat dibanding orang yang ditempatkan pada off-site. Karena untuk organisasi dengan penempatan fisik yang berlainan, off-site outsourcers ditempatkan dekat fasilitas yang jauh maka boleh jadi mampu menjawab dengan cepat dibanding orang yang ditempatkan di daerah pusat organisasi tersebut.
3. Kepekaan Data. Oleh karena ditujukan kepada kepekaan data dan privasi, beberapa organisasi mungkin segan untuk mengijinkan pihak eksternal memberikan gambaran *hard drives* dan melaksanakan tindakan lain yang menyediakan akses data. Sebagai contoh, suatu sistem yang berisi jejak dari suatu peristiwa mungkin juga berisi informasi pelayanan kesehatan, arsip keuangan, atau data sensitif lainnya, suatu organisasi mungkin lebih menyukai untuk menyimpan sistem tersebut dibawah kendali sendiri untuk melindungi kebebasan suatu data. Pada sisi lain, jika ada suatu privasi yang lebih ditujukan di dalam tim—sebagai contoh, suatu peristiwa yang dicurigai untuk melibatkan suatu anggota yang berhubungan dengan suatu peristiwa yang ditangani oleh tim—menggunakan pihak ketiga yang mandiri untuk melaksanakan tindakan analisa data.

*Incident handler* akan menyelenggarakan tugas analisa data yang dibutuhkan untuk dapat memiliki pengetahuan tentang prinsip analisa data yang kuat, prosedur, *tools* dan teknik, seperti halnya teknik dan *tools* yang bisa merahasiakan atau menghancurkan data. Hal ini menguntungkan *incident handler* untuk mempunyai keahlian di dalam keamanan suatu informasi dan teknis pokok yang spesifik, seperti sistem operasi yang digunakan, sistem file, aplikasi, dan protokol jaringan di dalam organisasi itu. Maka pengetahuan ini dapat memudahkan dalam pemberian respon yang lebih efektif dan lebih cepat ke suatu peristiwa. *Incident handler* juga memerlukan suatu pemahaman jaringan dan sistem umum yang luas sedemikian sehingga mereka dapat menentukan dengan cepat individu dan tim mana yang sesuai untuk menyediakan keahlian teknis untuk usaha analisa data tertentu, seperti memperoleh data untuk suatu aplikasi yang tidak biasa.

Individu yang menyelenggarakan analisa data dapat melaksanakan tugas yang lainnya juga. Sebagai contoh, *incident handler* dapat menyediakan latihan interaktif tentang analisa

data yang ditujukan kepada staff pendukung teknis, pengurus jaringan dan sistem dan para profesional IT lainnya. Contoh dari topik pelatihan mungkin meliputi suatu ikhtisar *tools* dan teknik analisa data, saran atas penggunaan alat tertentu dan tanda dari suatu serangan jenis baru. *Incident handler* dapat juga mempunyai sesi interaktif dengan kelompok para profesional IT untuk mendengar pemikiran mereka atas *tools* analisa data dan mengidentifikasi kekurangan potensi di dalam kemampuan analisa data yang ada.

Pada suatu peristiwa yang ditangani oleh tim, lebih dari satu anggota tim harus bisa melaksanakan masing-masing aktifitas analisa data yang khusus, sedemikian sehingga ketidakhadiran satu anggota regu mestinya tidak berdampak pada kemampuan tim tersebut. *Incident handler* dapat melatih satu sama lain tentang penggunaan *tools* analisa data dan topik mengenai cara dan teknis lain. Juga, latihan langsung dan kursus latihan analisa data dan IT eksternal dapat sangat menolong di dalam membangun dan memelihara ketrampilan. Mungkin juga merupakan pengaruh baik untuk anggota tim melihat demonstrasi tentang teknologi dan *tools* yang baru atau mencoba *tools* di dalam suatu laboratorium. Hal ini mungkin dapat sangat menolong untuk kebiasaan seseorang yang menangani suatu peristiwa dengan memperoleh dan menguji data dari alat seperti telepon selular dan PDA.

## 2.3 Interaksi dengan tim yang lain

Hal tersebut tidaklah mungkin bagi siapapun untuk menjadi lebih berpengalaman dalam tiap-tiap teknologi termasuk semua perangkat lunak yang digunakan di dalam suatu organisasi, maka individu yang melakukan tindakan analisa data harus bisa menggapai keluar individu dan tim lain di dalam organisasi mereka jika dibutuhkan untuk bantuan tambahan. Sebagai contoh, suatu peristiwa yang menyertakan server basis data tertentu mungkin ditangani lebih secara efisien jika pengurus database ada untuk menyediakan latar belakang informasi, masalah teknik jawaban dan menyediakan dokumentasi database dan material acuan lain. Maka, organisasi perlu memastikan bahwa para profesional IT sepanjang seluruh organisasi, terutama *incident handler* dan orang yang merespon suatu peristiwa, mereka perlu memahami tanggung-jawab dan peran mereka untuk analisa data, menerima pendidikan dan pelatihan atas data kebijakan terkait dengan analisa dan prosedur dan disiapkan untuk dapat bekerja sama dan membantu orang lain manakala teknologi yang mereka miliki merupakan suatu tanggung jawab untuk menjadi bagian dari suatu peristiwa atau kejadian lainnya.

Sebagai tambahan untuk para profesional dan *Incident handler*, selain di dalam suatu organisasi dapat juga mengambil bagian di dalam aktifitas analisa data dalam suatu kapasitas teknis yang lebih sedikit. Contohnya meliputi manajemen, penasehat hukum, sumber daya manusia, auditor dan staf keamanan fisik. Manajemen bertanggung jawab untuk mendukung kemampuan analisa data, meninjau ulang dan menyetujui data kebijakan terkait dengan analisa, dan tindakan analisa data tertentu yang disetujui ( misalnya mengambil suatu sistem *mission-critical* yang *off-line* 12 jam untuk memperoleh *hard drives*). Penasehat hukum perlu secara hati-hati meninjau ulang semua kebijakan analisa data dan prosedur tingkat tinggi dan mereka dapat menyediakan bimbingan tambahan manakala diperlukan untuk memastikan bahwa tindakan analisa data dilakukan dengan sah. Auditor dapat membantu menentukan dampak ekonomi dari suatu peristiwa, mencakup biaya dari aktifitas analisa data. Staf keamanan fisik dapat membantu dengan perolehan akses ke sistem. Jasa yang disediakan oleh tim ini dapat menjadi hal yang menguntungkan.

Untuk memudahkan komunikasi dalam tim, masing-masing tim perlu menandakan satu atau lebih poin-poin dari hubungan. Individu - individu ini untuk bertanggung jawab mengetahui keahlian dari tiap anggota tim dan penyelidikan secara langsung untuk bantuan kepada orang yang tepat. Organisasi perlu memelihara daftar informasi kontak yang dapat membantu tim yang tepat dalam mendapatkan referensi pada saat dibutuhkan. Suatu daftar dapat meliputi telepon kantor dan metode kontak pada saat keadaan darurat ( misalnya telepon selular) dimana keduanya bersifat standard.

## 2.4 Kebijakan

Organisasi perlu memastikan bahwa kebijakan mereka berisi statemen jelas yang menunjuk semua pertimbangan analisa data utama, seperti menyelenggarakan monitoring dan melaksanakan tinjauan ulang kebijakan analisa data yang reguler dan mengikuti prosedur. Pada level yang sudah tinggi, kebijakan perlu mengijinkan seseorang untuk diberi hak untuk memonitor jaringan dan sistem dan melaksanakan penyelidikan untuk pertimbangan yang sah sesuai dengan keadaan yang sebenarnya. Organisasi boleh juga mempunyai suatu kebijakan terpisah untuk orang yang menangani suatu masalah dan orang yang lain dengan peran menganalisa data yang akan menyediakan lebih terperinci peraturan untuk perilaku yang tepat. Orang seperti itu harus terbiasa dengan hal tesebut dan memahami kebijakan analisa data. Kebijakan diperbolehkan untuk sering di perbaharui, terutama sekali untuk organisasi

yang banyak memutar yurisdiksi, oleh karena perubahan peraturan dan hukum. Tentu saja, kebijakan analisa data organisasi harus konsisten dengan kebijakan organisasi yang lain. Bagian 2.4.1 sampai 2.4.3 mendiskusikan topik terkait dengan kebijakan secara lebih detil.

### 2.4.1 Penjelasan Tanggung-Jawab dan Peran

Kebijakan analisa data perlu dengan jelas menggambarkan tanggung-jawab dan peran dari semua orang menyelenggarakan atau membantu organisasi dengan aktifitas analisa data. Hal ini perlu meliputi tindakan yang dilakukan selama menangani peristiwa dan aktifitas pekerjaan rutin ( misalnya administrasi sistem, *network troubleshooting*). Kebijakan perlu meliputi semua tim internal yang dapat mengambil bagian di dalam usaha analisa data, seperti yang telah didaftarkan di dalam Bagian 2.3 dan organisasi eksternal seperti *outsourcers* dan *incident response organizations.*

Kebijakan perlu dengan jelas menandai siapa yang perlu dihubungi dari tim internal dan organisasi eksternal di bawah keadaan berbeda.

### 2.4.2 Menyediakan bimbingan untuk menggunakan alat analisa data

*Incident handler*, yaitu para profesional IT seperti pengurus jaringan dan sistem dan orang yang lain di dalam suatu organisasi dapat menggunakan teknik dan alat analisa data untuk berbagai pertimbangan. Walaupun teknologi mempunyai banyak manfaat, mereka dapat juga disalahgunakan secara kebetulan atau dengan sengaja untuk menyediakan akses untuk informasi yang tidak sah, atau untuk mengubah atau menghancurkan informasi. Juga penggunaan dari alat analisa data tertentu tidak mungkin dijamin dalam beberapa situasi—untuk contoh, suatu bagian kecil dari peristiwa yang mungkin tidak pantas menerima ratusan jam dalam usaha pengujian dan mendapatkan data.

Untuk memastikan bahwa alat yang digunakan layak dan sewajarnya, prosedur dan kebijakan suatu organisasi perlu dengan jelas menjelaskan apa yang dimaksud tindakan analisa data yang perlu dan tidak untuk dilakukan di bawah keadaan apapun. Sebagai contoh, suatu pengurus jaringan harus bisa memonitor komunikasi jaringan secara reguler untuk memecahkan permasalahan operasional, tetapi tidak untuk membaca *e-mail* para *user* kecuali jika diberi hak untuk melakukannya. Suatu *help desk agent* boleh jadi diijinkan untuk memonitor komunikasi jaringan untuk *user workstation* tertentu untuk dapat memperbaiki suatu masalah pada aplikasi, tetapi tidak diijinkan untuk melakukan monitoring di dalam jaringan lain. Para user individu boleh jadi dilarang untuk melakukan monitoring di jaringan

manapun di bawah situasi apapun. Prosedur dan Kebijakan perlu dengan jelas menggambarkan tindakan yang spesifik yang diijinkan dan terlarang untuk masing-masing peran yang bisa diterapkan dalam keadaan normal ( misalnya tugas-tugas khusus) dan keadaan khusus ( seperti menangani suatu peristiwa).

Prosedur dan Kebijakan perlu juga memberi petunjuk penggunaan teknik dan *tools* anti-forensik. Hal ini akan diuraikan dalam Bagian 4 sampai 7, perangkat lunak anti-forensik dirancang untuk merahasiakan atau menghancurkan data sehingga orang lain tidak bisa mengakses data tersebut. Ada banyak hal positif menggunakan untuk perangkat lunak anti-forensic, seperti pemindahan data dari komputer yang diharapkan untuk digunakan dan data pemindahan yang dirahasiakan oleh jaringan *browsers* untuk memelihara privasi seorang user. Bagaimanapun, hal seperti alat analisa data dan alat anti-forensik, kedua-duanya dapat digunakan untuk pertimbangan baik buruknya, maka organisasi perlu menetapkan siapa saja yang diijinkan untuk menggunakan *tools* tersebut dan pada situasi apa tools tersebut dapat digunakan .

Sebab alat analisa data dapat merekam informasi yang sensitif, prosedur dan kebijakan perlu juga menguraikan surat pengantar yang perlu untuk informasi itu. Untuk kearah sana perlu juga kebutuhan untuk menangani data yang terekspose tanpa sengaja dari informasi sensitif tersebut, seperti orang yang menangani suatu peristiwa melihat kata sandi atau informasi medis dari pasien.

### 2.4.3 Pendukung Analisa Data di dalam alur hidup dari Sistem informasi

Banyak peristiwa yang dapat ditangani lebih secara efisien dan secara efektif jika pertimbangan analisa data telah disatukan ke dalam alur hidup dari Sistem informasi. Contoh pertimbangan yang seperti itu adalah sebagai berikut:

1. Menyelenggarakan *reguler backups* pada sistem dan memelihara *backups* yang sebelumnya untuk suatu periode waktu spesifik
2. Memungkinkan auditing pada *workstation*, server, dan alat jaringan
3. Penyampaian arsip audit untuk menjamin log server yang diutamakan
4. Konfigurasi aplikasi - aplikasi *mission-critical* untuk melaksanakan *auditing*, termasuk yang merekam semua usaha pengesahan
5. Pemeliharaan suatu basis data dari banyak file untuk file tentang sistem operasi dan aplikasi penyebaran umum
6. Pemeliharaan arsip ( misalnya *baselines*) tentang bentuk wujud sistem dan jaringan

7. Penetapan kebijakan penyimpanan data yang mendukung tinjauan ulang aktifitas jaringan dan sistem historis.

Kebanyakan pertimbangan ini adalah perluasan dari ketentuan yang ada di dalam kebijakan dan prosedur, sehingga mereka ditetapkan secara khusus di dalam dokumen individu yang relevan sebagai ganti suatu kebijakan yang diutamakan pada analisa data.

## 2.5 Prosedur

Seperti yang telah disebutkan di dalam Bagian 2.4, suatu organisasi perlu menciptakan dan memelihara prosedur untuk dapat menyelenggarakan tugas analisa data, berdasar pada kebijakan organisasi, menanggapi suatu peristiwa penyusunan model kepegawaian dan tim lain yang mengidentifikasikan seperti peserta di dalam aktifitas analisa data. Sekalipun aktifitas analisa data dilakukan oleh pihak eksternal, staff internal organisasi masih saling berhubungan dengan mereka dan mengambil bagian sampai taraf tertentu di dalam aktifitas analisa, seperti memberitahu pihak eksternal untuk suatu kebutuhan akan bantuan dan memberikan fisik atau akses logis ke sistem. Staff internal perlu bekerja dengan dengan bagian eksternal untuk dapat memastikan bahwa prosedur organisasi dan kebijakan dapat dipahami dan diikuti.

Prosedur perlu meliputi metodologi umum untuk menyelidiki suatu peristiwa yang menggunakan teknik analisa data, karena tidak mungkin untuk mengembangkan prosedur menyeluruh yang dikhususkan ke tiap-tiap situasi yang mungkin. Bagaimanapun, organisasi perlu mempertimbangkan tentang *step-by-step* prosedur pengembangan untuk melakukan tugas – tugas yang rutin seperti *imaging* suatu *hard-disk*, menangkap dan merekam informasi yang bersifat *volatile* dari sistem. Tujuannya adalah untuk prosedur dalam memfasilitasi secara konsisten, efektif dan tindakan analisa data akurat. Tentu saja, prosedur perlu juga konsisten dengan kebijakan organisasi dan semua ketentuan hukum yang bisa diterapkan. Maka, organisasi perlu meliputi tenaga ahli dan penasehat hukum di dalam pengembangan prosedur sebagai ukuran jaminan mutu. Manajemen perlu juga dilibatkan di dalam pengembangan prosedur, yang terutama sekali untuk menjamin bahwa semua pengambilan keputusan poin-poin utama didokumentasikan dan tindakan yang sesuai digambarkan, sedemikian sehingga keputusan dapat dibuat secara konsisten.

Adalah suatu hal penting juga untuk memelihara prosedur sedemikian sehingga dapat merupakan prosedur yang akurat. Manajemen perlu menentukan sesering apakah prosedur harus ditinjau ulang (biasanya sedikitnya tiap tahun). Tinjauan ulang perlu juga diselenggarakan manakala perubahan penting dibuat untuk kebijakan dan prosedur tim. Manakala suatu prosedur diperbaharui, versi yang sebelumnya harus disimpan untuk masa depan karena mungkin saja dapat digunakan di dalam cara bekerja yang legal. Tinjauan ulang prosedur harus dilakukan oleh tim yang sama yang mengambil bagian di dalam pembuatan prosedur. Sebagai tambahan, organisasi mungkin juga memilih untuk melakukan latihan yang membantu ke arah mengesahkan ketelitian dari prosedur tertentu.

## 2.6 Rekomendasi

Kunci rekomendasi diperkenalkan di dalam bagian ini untuk pengaturan suatu kemampuan analisa data yang akan dirangkum seperti dibawah ini :

1. Organisasi perlu mempunyai beberapa kemampuan untuk melaksanakan analisa data jaringan dan komputer. Analisa data dapat membantu berbagai tugas di dalam suatu organisasi, termasuk merekonstruksi peristiwa keamanan komputer, penanganan permasalahan operasional dan pemulihan dari kerusakan sistem yang terjadi secara kebetulan.
2. Organisasi perlu menentukan bagian mana yang harus menangani masing-masing aspek dari analisa data. Kebanyakan organisasi bersandar pada suatu kombinasi dari staff mereka sendiri dan pihak eksternal untuk melaksanakan tugas analisa data. Organisasi perlu memutuskan pihak mana yang perlu memperhatikan tugas berdasar pada kemampuan dan ketrampilan, biaya, waktu tanggapan, dan kepekaan data.
3. Tim yang menangani suatu peristiwa perlu mempunyai kemampuan analisa data sempurna. Lebih dari satu anggota regu harus bisa melaksanakan masing-masing aktifitas analisa data khusus. IT dan latihan langsung dan kursus latihan analisa data dapat sangat menolong di dalam membangun dan memelihara ketrampilan, seperti dapat mendemonstrasikan teknologi dan *tools* baru.
4. Banyak tim di dalam suatu organisasi yang perlu mengambil bagian dalam analisa data. Individu yang melakukan tindakan analisa data harus bisa menggapai ke luar ke individu dan tim lain di dalam suatu organisasi jika dibutuhkan untuk bantuan tambahan. Contohya tim yang meliputi para profesional IT, manajemen, penasehat hukum, auditor, dan staf

keamanan fisik. Anggota dari tim ini perlu memahami tanggung-jawab dan peran mereka untuk analisa data, menerima pendidikan dan pelatihan atas analisa data yang berhubungan dengan kebijakan dan prosedur dan mempersiapkan diri untuk dapat bekerja sama dengan orang lain serta dapat membantu orang lain atas tindakan analisa data.

5. Pertimbangan analisa data harus dengan jelas ditujukan di dalam kebijakan. Pada suatu tingkatan yang tinggi, kebijakan perlu mengijinkan seseorang untuk diberi hak untuk memonitor jaringan dan sistem dan melaksanakan penyelidikan untuk pertimbangan yang sah sesuai dengan keadaan. Organisasi juga boleh mempunyai suatu kebijakan analisa data terpisah untuk *incident handler* dan orang yang lain dengan peran menganalisa data yang telah menyediakan lebih terperinci peraturan untuk perilaku yang sesuai. Semua orang yang mungkin dipanggil untuk membantu usaha analisa data manapun harus terbiasa dengan kebijakan analisa dan memahaminya. Pertimbangan kebijakan tambahan sebagai berikut:
   a. Kebijakan analisa data perlu dengan jelas menggambarkan tanggung-jawab dan peran dari semua orang yang melakukan atau membantu dengan aktifitas analisa data organisasi. Kebijakan perlu meliputi semua pihak internal dan eksternal yang mungkin untuk dilibatkan dan hal tersebut perlu dengan jelas menandai siapa yang perlu dihubungi dari suatu pihak dengan kondisi yang berbeda - beda .
   b. Prosedur dan kebijakan suatu organisasi perlu dengan jelas menjelaskan apa yang dimaksud dengan tindakan analisa data yang perlu dan yang tidak perlu dilakukan di kondisi tertentu ataupun kondisi normal dan juga memberi petunjuk penggunaan tentang teknik dan *tools* anti-forensik. Prosedur dan kebijakan perlu juga memberi petunjuk penanganan tentang informasi sensitif yang secara tidak sengaja terekspose.
   c. Bersama pertimbangan analisa data di dalam siklus hidup sistem informasi dapat mendorong kearah penanganan yang lebih efektif dan efisien dari banyak peristiwa.

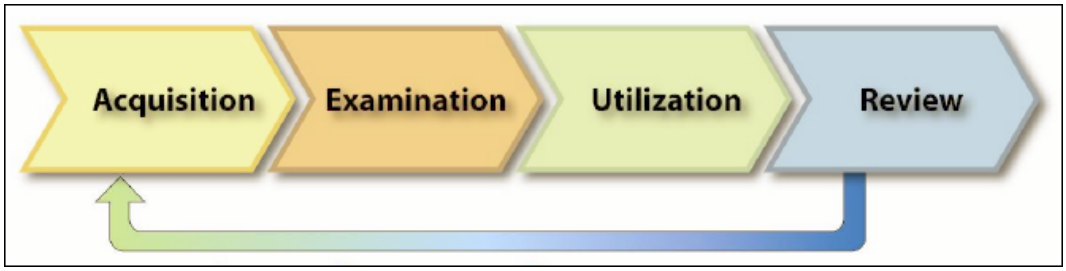
6. Organisasi perlu menciptakan dan memelihara prosedur untuk menyelenggarakan tugas analisa data. Prosedur perlu meliputi metodologi umum untuk menyelidiki suatu peristiwa yang menggunakan teknik analisa data dan mungkin prosedur *step-by-step* untuk melakukan tugas rutin. Prosedur harus ditinjau secara teratur dan dirawat sedemikian sehingga mereka akurat

# Pelaksanaan Proses Analisa Data

# 3

Tujuan umum dalam menyelenggarakan analisa data jaringan dan komputer adalah untuk memperoleh suatu pemahaman lebih baik dari suatu peristiwa yang menarik dimana kita dapat menemukan dan meneliti fakta yang menyertai peristiwa tersebut. Seperti yang diuraikan di dalam Bagian 2.1, analisa data mungkin diperlukan di dalam banyak situasi yang berbeda, seperti peristiwa penanganan yang efektif terhadap *malware* dan permasalahan operasional yang tidak biasa. Dengan mengabaikan kebutuhan, pengujian dan darimana di dapatnya data maka harus dilakukan dengan menggunakan proses yang terdiri dari empat fase sama yang ditunjukkan di dalam Gambar 3-1. Detil yang tepat dari langkah-langkah ini boleh ditukar berdasarkan kebutuhan yang spesifik.

Bagian ini menguraikan tahap dari proses analisa data: bagaimana cara didapatnya data, pengujian, pemanfaatan, dan tinjauan ulang. Sepanjang tahap bagaimana cara didapatnya data, data berhubungan dengan suatu peristiwa spesifik yang sudah dikenali, dikumpulkan, dan dilindungi. Tahap yang kedua , pengujian, adalah waktu dimana teknik dan *tools* yang sesuai dengan jenis data yang telah dikumpulkan sepanjang tahap yang pertama data dieksekusi untuk mengidentifikasi dan meneliti informasi yang relevan dari data yang diperoleh tersebut. Tahap yang kedua boleh menggunakan suatu kombinasi dari proses manual dan otomatis *tools*. Tahap yang berikutnya, pemanfaatan, yaitu menyiapkan dan mempresentasikan hasil dari proses pengujian dalam suatu format yang dapat dengan mudah berasimilasi oleh *audience*. Hasilnya mungkin perlu untuk dipresentasikan ke *incident handlers*, para profesional IT, *end user*, manajemen atau orang lain. Dalam banyak kesempatan, tahap pemanfaatan merupakan hal yang sepele sebab orang yang melakukan analisa juga adalah konsumen yang menggunakan hasil tersebut. Tahap akhir melibatkan praktek dan proses meninjau ulang dalam konteks peristiwa yang sekarang untuk mengidentifikasi kekurangan kebijakan, kesalahan prosedur, dan permasalahan lain yang perlu untuk diperbaiki. Belajar mempelajari sepanjang tahap tinjauan ulang harus disatukan ke dalam usaha analisa data di masa depan.




Gambar 3.1. Analisa Data Jaringan dan Komputer Memproses

## 3.1 Memperoleh Data

Langkah pertama di dalam proses ini adalah memperoleh suatu data dan mengidentifikasi sumber potensi data tersebut. Bagian 3.1.1 menggambarkan variasi sumber data dan mendiskusikan tindakan organisasi yang dapat mendukung untuk mendapatkan kumpulan data secara berkelanjutan. Bagian 3.1.2 menguraikan langkah-langkah yang direkomendasikan untuk pengumpulan data. Bagian 3.1.3 mendiskusikan pertimbangan dalam menanggapi peristiwa, menekankan kebutuhan untuk menimbang nilai tentang data yang diperoleh terhadap biaya-biaya dan dampak terhadap organisasi dilihat dari proses didapatnya data tersebut.

### 3.1.1 Sumber Data yang Mungkin

Meluasnya penggunaan tentang teknologi digital untuk profesional dan pertimbangan pribadi telah mendorong suatu sumber data untuk berlimpah. Sumber data yang umum dan jelas nyata adalah komputer desktop, server, *network storage devices*, dan laptop. Sistem ini yang secara khusus mempunyai *internal drives* yang menerima media, seperti CD dan DVD, dan juga mempunyai beberapa jenis port (misalnya Serial Universal Bus [ USB], Firewire, Personal Computer Memory Card International Association [ PCMCIA]) untuk data eksternal.

Media penyimpanan dan alat dapat dipasang. Contohnya adalah tentang format penyimpanan eksternal yang mungkin saja sumber datanya adalah *thumb drives*, memori dan *flash card*, *optical discs* dan *magnetic disks.* Sistem komputer standard juga berisi data yang bersifat *volatile* yang ada tersedia untuk sementara ( yaitu, sampai sistem dimatikan atau *rebooted*). Sebagai tambahan terhadap alat terkait dengan komputer, banyak jenis alat digital bersifat *portable* yang berisi data. Alat ini meliputi PDA, telepon selular, kamera digital, perekam digital dan *audio players.*

Sumber data yang sering ditempatkan ditempat lain. Sebagai contoh, seperti yang diuraikan di dalam Bagian 6 dan 7, pada umumnya banyak sumber informasi di dalam suatu organisasi yang mengenai pemakaian aplikasi dan aktifitas jaringan.Informasi boleh juga direkam oleh organisasi lain, seperti *logs* aktifitas jaringan untuk suatu Internet Service Provider ( ISP). Analis perlu juga untuk ingat akan pemilik dari tiap sumber data dan efeknya adalah memungkinkan memperoleh data. Sebagai contoh, mengambil salinan dari arsip ISP secara khusus memerlukan suatu *court order*. Analis perlu juga menyadari kebijakan organisasi dan pertimbangan tentang undang-undang mengenai hak milik secara eksternal yang dimiliki di fasilitas organisasi, seperti suatu laptop pribadi sebagai pekerja atau *contractor's laptop*. Situasi ini dapat menjadi lebih rumit lagi jika penempatan di luar kendali organisasi itu dilibatkan, seperti peristiwa yang menyertakan suatu komputer pada suatu *telecommuter* kantor pusat. Kadang-kadang masalah seperti itu dapat mengakibatkan tidak diperolehnya suatu sumber data primer; analis harus sadar akan sumber data pengganti yang dapat berisi beberapa atau semua data yang sama dan menggunakan sumber itu sebagai ganti sumber yang tidak dapat dicapai.

Organisasi dapat mengambil ukuran proaktif berkelanjutan untuk mengumpulkan data bermanfaat. Sebagai contoh, seperti yang diuraikan di dalam Bagian 5.1.1, kebanyakan sistem operasi dapat diatur untuk diaudit dan record yang tertentu jenis berdasarkan peristiwa, seperti perubahan kebijakan keamanan dan usaha pengesahan, sebagai bagian dari operasi normal. Arsip audit dapat menyediakan informasi berharga, mencakup waktu suatu peristiwa terjadi dan asal dari event. Tindakan lain yang sangat membantu dalam menerapkan *centralized logging*, yang berarti sistem yang sudah tentu dan dimana salinan aplikasi meneruskan log tersebut untuk mengamankan pusat server dari log. *Centralized logging* mencegah para user tidak sah dari perusakkan log dan teknik yang dimanfaatkan untuk menghalangi suatu analisa. Menyelenggarakan *reguler backups* dari sistem mengijinkan analis untuk memandang muatan dari sistem sebagai mereka pada situasi tertentu. Juga seperti yang diuraikan di dalam bagian 6 dan 7, kendali *monitoring* keamanan seperti perangkat lunak yang mendeteksi gangguan, perangkat lunak antivirus dan pendeteksi spyware dan manfaat pemindahan yang dapat menghasilkan log yang menunjukkan kapan dan bagaimana caranya suatu gangguan atau serangan mengambil alih suatu tempat.

Ukuran pengumpulan data Proaktif yang lain adalah *keystroke monitoring*, seperti *record* pemakaian keyboard dari suatu sistem tertentu.Walaupun hal tersebut dapat

menyediakan suatu *record* aktifitas yang berharga, hal tersebut juga dapat menjadi pelanggaran dari privasi kecuali jika para user disarankan bahwa *monitoring* yang seperti itu mungkin dilakukan melalui kebijakan organisasi dan *login banners*. Kebanyakan organisasi tidak mempekerjakan *keystroke monitoring* kecuali ketika pengumpulan informasi tambahan atas suatu peristiwa yang dicurigai. Otoritas untuk melakukan *monitoring* seperti itu harus dibahas dengan penasehat hukum dan didokumentasikan dengan jelas di dalam itu kebijakan organisasi.

### 3.1.2 Mengumpulkan Data

Setelah mengidentifikasi sumber data yang potensial, analis harus memperoleh data dari sumber itu. Didapatnya data harus dilakukan dengan menggunakan proses tiga langkah: mengembangkan suatu rencana untuk memperoleh data, memperoleh data dan pembuktian integritas dari data yang diperoleh. Walaupun materi berikut menyediakan suatu ikhtisar yang menyangkut ketiga langkah tersebut, detil dari spesifik yang ada di langkah-langkah 2 dan 3 dapat ditukar - tukar berdasarkan atas jenis data yang sedang diperoleh. Bagian 4.2, 5.2, 6.3 dan 7.3 menyediakan lebih terperinci penjelasan dalam memperoleh dan membuktikan secara berturut-turut tentang integritas file data, sistem operasi data *network traffic data* dan data aplikasi.

1. Mengembangkan suatu rencana untuk memperoleh data. Mengembangkan suatu rencana adalah suatu langkah pertama yang penting dalam banyak kasus sebab ada berbagai sumber data yang berpotensi. Analis perlu menciptakan suatu rencana tentang sumber yang diprioritaskan, menetapkan pesanan di mana data harus diperoleh. Faktor penting untuk prioritisasi meliputi yang berikut ini:
    a. Nilai yang Mungkin. Didasarkan pada pemahaman para analis akan situasi dan pengalaman sebelumnya di dalam situasi yang serupa, analis harus bisa menaksir nilai yang mungkin bersifat relatif dari tiap sumber data yang potensial.
    b. *Volatility*. Data yang bersifat *volatile* mengacu pada data dalam suatu tempat yang berada pada sistem dimana data tersebut akan hilang setelah komputer dimatikan. Data *volatile* juga dapat hilang dalam kaitan dengan tindakan lain yang dilakukan terhadap sistem tersebut. Dalam banyak kasus, perolehan data yang bersifat *volatile* harus diberikan prioritas daripada data yang *non volatile*. Bagaimanapun, data yang *non volatile* juga dapat bersifat agak dinamis secara alami, seperti *log files* dimana akan melakukan *overwritten* dari ketika peristiwa baru terjadi.

c. Jumlah usaha yang diperlukan. Jumlah usaha yang diperlukan untuk memperoleh sumber data berbeda yang dapat bertukar-tukar secara luas. Usaha melibatkan tidak hanya waktu yang digunakan oleh orang yang lain dan analis di dalam organisasi ( termasuk penasehat hukum), tetapi juga meliputi biaya jasa dan peralatan ( seperti., tenaga ahli dari luar). Sebagai contoh, memperoleh data dari suatu *network router* mungkin akan memerlukan sangat sedikit usaha dibanding memperoleh data dari suatu ISP.

Dengan mempertimbangkan tiga faktor ini untuk masing-masing sumber data yang potensial, analis dapat membuat keputusan yang diberitahukan mengenai prioritisasi didapatnya sumber data, seperti halnya menentukan sumber data yang mana untuk memperoleh. Dalam beberapa hal, sangat banyak sumber data yang mungkin tidak praktis untuk memperoleh semua data tersebut. Organisasi perlu secara hati-hati mempertimbangkan kompleksitas dalam memprioritaskan didapatnya sumber data dan mengembangkan rencana tertulis, petunjuk dan prosedur yang dapat membantu analis di dalam menyelenggarakan prioritisasi secara efektif.

2. Memperoleh data. Jika data belum diperoleh oleh *security tools, analysis tools* atau alat-alat lain, proses yang umum untuk memperoleh data melibatkan penggunaan suatu *toolkit* dipercayai untuk mengumpulkan data *volatile* dan menyalin sumber data *non-volatile* untuk dapat mengumpulkan data tersebut. Didapatnya data dapat dilakukan yang manapun di tempat itu atau pada suatu jaringan. Walaupun umumnya disepakati untuk lebih baik memperoleh data di tempat itu sebab ada kendali lebih besar atas data dan sistem, pengumpulan data lokal tidaklah selalu mungkin (seperti, sistem dalam ruang yang dikunci, sistem di dalam lokasi yang lain ). Manakala memperoleh data yang ada pada suatu jaringan, keputusan harus dibuat berdasarkan jenis data yang akan dikumpulkan dan jumlah usaha yang digunakan. Sebagai contoh, hal tersebut mungkin saja diperlukan untuk memperoleh data dari beberapa sistem melalui koneksi jaringan berbeda atau hal tersebut cukup untuk meng*copy* suatu volume yang logis hanya dari satu sistem.

3. Memverifikasi integritas dari data. Setelah data telah diperoleh, integritasnya harus dibuktikan. Verifikasi integritas data secara khusus terdiri dari *penggunaan tools* untuk

menghitung isi suatu pesan yang asli dan meng*copy* data dan membandingkannya untuk meyakinkan bahwa data tersebut adalah sama.

**Sebelum analis memulai untuk memperoleh data manapun, suatu keputusan harus dibuat oleh manajemen atau analis ( sesuai dengan kebijakan organisasi dan penasehat hukum) atas kebutuhan untuk memperoleh dan memelihara bukti dengan cara yang mendukung penggunaannya dalam cara bekerja teratur pada pihak internal atau legal pada masa depan. Dalam situasi yang sedemikian, prosedur jaringan dan komputer forensik harus diikuti sebagai ganti semakin sedikitnya prosedur analisa data formal dibahas di dalam pemandu ini.**

Untuk membantu analis dengan *acquisition*, sumber daya yang perlu harus disiapkan terlebih dahulu, seperti *analyst workstations, backup devices,* dan media kosong.

### 3.1.3 Pertimbangan Respon terhadap Peristiwa

Manakala menyelenggarakan analisa data selama *incident response*, suatu pertimbangan yang penting adalah bagaimana dan kapan suatu peristiwa harus dimasukkan. Mengisolasi sistem yang bersangkutan dari pengaruh eksternal mungkin perlu untuk memelihara integritas data. Dalam banyak kasus, analis perlu bekerja dengan tim yang menanggapi suatu kejadian untuk membuat suatu keputusan *containment* (seperti, pemutusan kabel jaringan, *unplugging power*, meningkatnya ukuran keamanan fisik, menutup *host*). Keputusan ini harus didasarkan pada kebijakan yang ada dan prosedur mengenai peristiwa *containment*, seperti halnya penilaian tim dari resiko yang diajukan oleh peristiwa, sedemikian sehingga strategi kombinasi atau *containment* strategi yang terrpilih cukup mengurangi resiko selagi pemeliharaan integritas data yang mungkin kapan saja.

Organisasi perlu juga mempertimbangkan dampak yang dahulu dari berbagai strategi *containment* yang mungkin mempunyai kemampuan dari organisasi untuk beroperasi secara efektif. Sebagai contoh, mengira suatu sistem kritis *offline* beberapa jam untuk memperoleh gambaran disk dan data lain dapat mempengaruhi secara kurang baik kemampuan dari organisasi untuk melaksanakan operasi yang diperlukan. Kekurangan waktu penting dapat mengakibatkan kerugian moneter substansiil kepada organisasi itu. Oleh karena itu, kepedulian harus diambil untuk memperkecil gangguan kepada suatu operasi sebuah organisasi.

Satu langkah sering diambil untuk mengetahui suatu peristiwa yang akan mengamankan garis pertahanan di sekitar komputer dan membatasi akses untuk memberi hak kepada personil. Juga hal tersebut kadang-kadang sangat menolong untuk menciptakan daftar semua para user yang mempunyai akses ke komputer, sebab mereka mungkin mampu menyediakan kata sandi atau informasi tentang dimana data spesifik ditempatkan. Jika komputer dihubungkan untuk suatu jaringan, memutuskan kabel jaringan yang terkait dengan komputer dapat mencegah para *remote users* dari memodifikasi data komputer itu. Jika komputer menggunakan suatu koneksi jaringan tanpa kabel, melepaskan *adapte*r jaringan yang eksternal dari komputer atau melumpuhkan *adapter* jaringan internal mungkin dapat digunakan untuk memotong koneksi jaringan. Jika tidak ada pilihan yang mungkin, dengan mematikan akses poin jaringan tanpa kabel yang dihubungkan komputer tersebut maka akan mendapatkan hasil yang sama. Bagaimanapun, hal tersebut dapat mencegah para user diluar lingkup penyelidikan dari melakukan rutinitas mereka sehari-hari. Juga, hal tersebut dapat lebih dari satu akses poin dalam jangkauan suatu komputer. Beberapa orang yang mengadaptasikan jaringan tanpa kabel secara otomatis mencoba untuk menghubungkan ke akses poin lain manakala titik akses yang utama tak tersedia, jadi mengisi suatu peristiwa dengan cara ini bisa melibatkan pemutusan beberapa poin akses.

## 3.2 Pengujian

Setelah data telah diperoleh, tahap yang berikutnya adalah untuk menguji data, yang mana terdiri dari mengidentifikasi, mengumpulkan dan mengorganisir bagian informasi yang relevan dari data yang diperoleh. Tahap ini juga dapat melibatkan pemotongan atau pengurangan sistem operasi atau menonjolkan aplikasi yang mengaburkan kode dan data, seperti data yang dikompres, *encryption* dan mekanisme akses kontrol. Sebagai contoh, diperolehnya suatu *harddrive* boleh berisi ratusan ribu file data; mengidentifikasi file data yang berisi informasi yang menarik, mencakup informasi yang tersembunyi sampai file yang dikompres dan kontrol akses, bisa merupakan suatu tugas yang menakutkan. Apalagi, file data yang penting dapat berisi informasi asing tambahan yang harus disaring. Sebagai contoh, *firewall log* kemarin dapat menjaga berjuta-juta arsip, tetapi hanya sebanyak lima arsip yang dihubungkan dengan peristiwa tertentu yang menarik .

Kebetulan, berbagai teknik dan *tools* dapat digunakan untuk mengurangi jumlah data yang harus sampai disaring. Teks dan pola pencarian dapat digunakan untuk mengidentifikasi

data bersangkutan, seperti temuan dokumen yang menyebutkan orang atau pokok tertentu atau mengidentifikasi masukan *log* pada *e-mail* untuk alamat *e-mail* tertentu. Teknik lain yang sangat menolong adalah untuk menggunakan suatu alat yang dapat menentukan jenis muatan dari tiap file data, seperti teks, grafik, musik atau suatu arsip file yang dikompres. Pengetahuan jenis file data dapat digunakan untuk mengidentifikasi file yang pantas menerima studi lebih lanjut, seperti halnya untuk meniadakan file yang tidak menarik ke proses pengujian. Ada juga database yang berisi informasi pada file yang dikenal, yang mana dapat juga digunakan untuk memasukkan atau meniadakan file dari pertimbangan lebih lanjut. Informasi spesifik atas teknik dan tools pengujian diperkenalkan dalam Bagian 4.3, 5.3, 6.4, dan 7.4.

Ketika sekali saja informasi yang relevan telah disadap, analis perlu belajar data untuk menarik kesimpulan dari data itu. Analis perlu mengikuti suatu pendekatan metodis untuk menarik kesimpulan berdasar pada data yang tersedia atau menentukan tidak ada sekalipun kesimpulan yang dapat ditarik. Analisa perlu meliputi mengidentifikasi orang, tempat, materi dan peristiwa, dan menentukan bagaimana mereka terkait sedemikian sehingga suatu kesimpulan dapat dicapai. Sering kali, hal ini akan meliputi penghubungan data antar berbagai sumber. Sebagai contoh, suatu *network IDS log* dapat menghubungkan suatu peristiwa ke suatu *hos*t, *host audit logs* dapat menghubungkan peristiwa ke rekening spesifik seorang pemakai, dan *host IDS log* dapat menandai adanya tindakan yang dilakukan pemakai. *Tools* seperti perangkat lunak *security event management* dan *centralized logging* dapat memudahkan proses ini dengan secara otomatis mengumpulkan dan menghubungkan data itu. Juga, membandingkan karakteristik sistem untuk mengenal *baselines* yang dapat mengidentifikasi berbagai jenis perubahan yang dibuat kepada sistem tersebut. Bagian 8 menguraikan analisa memproses secara lebih detil.

## 3.3 Pemanfaatan

Pemanfaatan data adalah proses menyiapkan dan mempresentasikan informasi yang diakibatkan oleh tahap pengujian. Banyak faktor mempengaruhi pemanfaatan data, mencakup berikut ini:

1. Pengurangan Data. Mengurangi data untuk menyajikan fakta saja yang perlu untuk membantu orang yang sesuai ke arah yang pasti dari suatu keseluruhan atas pemahaman dari apa yang telah terjadi dan mungkin menandai apa yang perlu

dilaksanakan untuk melakukan koreksi atau memodifikasi suatu isu. Jika suatu analis sedang mengidentifikasi file yang dikirim lewat email oleh suatu virus ke komputer yang kemudian menjadi tersebar dalam komputer, orang yang menggunakan komputer yang ingin melihat nama dari file yang dimasalahkan akan mungkin tidak tertarik pada *display* mencakup *e-mails* yang dikirim ke *account* yang sama.

2. Penjelasan Alternatif. Manakala informasi mengenai suatu peristiwa tidak sempurna, maka hal terebut tidaklah mungkin untuk mengidentifikasi suatu penjelasan pasti seperti apa yang terjadi . Manakala suatu peristiwa mempunyai dua atau lebih penjelasan masuk akal, masing-masing harus diberi hak pertimbangan di dalam proses pemanfaatan data.
3. Pertimbangan *Audience*. Mengetahui pendengar untuk informasi atau data mana yang disimpan yang akan jadi hal penting untuk ditunjukkan. *System administrator* mungkin ingin melihat *network traffic* dan terkait statistik dengan rincian yang baik. Manajemen senior mungkin sederhananya ingin suatu ikhtisar tingkat tinggi dari apa yang terjadi, seperti suatu penyajian visual yang disederhanakan tentang bagaimana serangan terjadi, dan apa yang perlu dilaksanakan untuk mencegah peristiwa yang serupa.
4. *Actionable Information.* Pemanfaatan juga meliputi *Actionable Information* dieperoleh dari data yang boleh mengijinkan suatu analis untuk memperoleh sumber informasi baru. Sebagai contoh, daftar kontak mungkin saja disimpan dari data yang dapat mendorong kearah informasi tambahan tentang suatu peristiwa. Juga, informasi yang mungkin diperoleh itu bisa mencegah peristiwa pada masa depan, seperti suatu *backdoor* pada suatu sistem yang bisa digunakan untuk serangan di masa depan atau *worm* yang dijadwalkan untuk memulai penyebaran pada suatu waktu tertentu.

## 3.4 Tinjauan ulang

Analis perlu secara terus-menerus meninjau ulang proses tersebut dan praktek dalam konteks tugas sekarang untuk membantu mengidentifikasi kekurangan kebijakan, kesalahan prosedur, dan isu lain yang boleh perlu untuk diperbaiki. Penyegaran ketrampilan yang berkala sampai *coursework*, pengalaman *on-the-job* dan bantuan sumber akademis akan memastikan bahwa orang yang melakukan akan analisa data melangkah dengan cepat mengubah tanggung-jawab pekerjaan dan teknologi tanggung-jawab. Tinjauan ulang

kebijakan yang berkala dan prosedur juga membantu memastikan organisasi sekarang tetap dengan trend teknologi dan berubah dalam hukum.

Banyak tanggapan tim terhadap suatu peristiwa memegang tinjauan ulang formal setelah masing-masing peristiwa utama. Tinjauan ulang seperti itu cenderung meliputi pertimbangan serius tentang segala peningkatan yang mungkin untuk proses dan prosedur, dan sedikitnya secara khusus beberapa perubahan disetujui dan diterapkan setelah masing-masing tinjauan ulang. Sebagai contoh, banyak organisasi menemukannya *resource-intensive* untuk memelihara daftar personil yang sekarang untuk dihubungi mengenai masing-masing jenis peristiwa berbeda yang dapat terjadi. Sekali perubahan prosedur dan proses diterapkan, semua anggota tim harus diberitahukan menyangkut perubahan dan seringkali diingatkan tentang prosedur yang sesuai untuk diikuti. Tim yang secara khusus mempunyai mekanisme formal untuk perubahan jalan dan mengidentifikasi versi yang sekarang dari tiap proses dan dokumen prosedur. Sebagai tambahan, banyak tim mempunyai poster atau dokumen yang terlihat diatas pintu atau dinding yang mengingatkan tim menyangkut langkah-langkah penting yang akan diambil, sehingga semua orang secara konstan diingatkan bagaimana suatu hal dianggap benar untuk dilaksanakan.

## 3.5 Rekomendasi

Kunci dari banyak rekomendasi untuk proses analisa data yang diperkenalkan di dalam bagian ini diringkas sebagai berikut.

1. Organisasi perlu melaksanakan analisa data dengan menggunakan suatu proses yang konsisten. Proses analisa data diperkenalkan di dalam panduan ini menggunakan suatu proses dengan empat fase: perolehan data, pengujian, pemanfaatan, dan tinjauan ulang. Rincian yang tepat dari tahap tersebut dapat diacak berdasarkan pada kebutuhan akan analisa data.
2. Analis harus sadar akan cakupan dari sumber data mungkin. Analis harus bisa mensurvei suatu area fisik dan mengenali sumber data yang mungkin. Analis perlu juga berpikir tentang sumber data mungkin ditempatkan di lokasi lain di dalam suatu organisasi dan di luar organisasi itu . Analis harus disiapkan untuk menggunakan sumber data alternatif jika hal tersebut tidak memungkinkan untuk memperoleh data dari suatu sumber utama.

3. Organisasi harus proaktif dalam pengumpulan data yang bermanfaat. *Configuring auditing* pada sistem operasi, menerapkan *centralized logging*, melakukan sistem *reguler backups*, dan menggunakan *security monitoring controls* dapat menghasilkan semua sumber data untuk usaha analisa data di masa depan.
4. Analis perlu melaksanakan proses mendapatkan data yang menggunakan suatu proses standar. Langkah-langkah yang diusulkan adalah mengembangkan suatu rencana untuk perolehan data, memperoleh data dan membuktikan integritas dari data. Analis perlu menciptakan suatu rencana yang memprioritaskan sumber data, menetapkan order dimana data harus diperoleh, didasarkan pada nilai yang mungkin dari data, sifat volatil dari data dan jumlah usaha yang dibutuhkan.
5. Analis perlu menggunakan suatu pendekatan yang metodis. Pondasi bagi analisa data jaringan dan komputer adalah menggunakan suatu pendekatan yang metodis untuk menggambarkan kesimpulan berdasar pada data yang tersedia atau menentukan bahwa tidak ada kesimpulan sekalipun digambarkan.
6. Analis perlu meninjau ulang proses dan praktek mereka. Tinjauan ulang tentang tindakan analisa data terbaru dan sekarang dapat membantu mengidentifikasi kekurangan kebijakan, kesalahan prosedur, dan isu lain yang mungkin perlu untuk diperbaiki, seperti halnya memastikan bahwa organisasi sekarang tetap dengan kecenderungan di dalam teknologi dan berubah karena hukum.

# Penggunaan Data dari File Data

# 4

Suatu file data ( atau sering disebut dengan file) adalah suatu kumpulan dari informasi yang secara logika dikelompokkan ke dalam kesatuan tunggal dan disesuaikan oleh suatu nama yang unik, seperti suatu nama file. Suatu file dapat menjadi banyak tipe dari suatu data, mencakup suatu dokumen, suatu gambaran, suatu video atau suatu aplikasi. Pengujian media komputer yang sukses adalah bergantung pada kemampuan memperoleh, menyadap, dan meneliti file yang berada pada media itu. Bagian ini mulai dengan suatu ikhtisar menyangkut tipe – tipe media yang paling umum dan *filesystems*—metode untuk menamai, menyimpan, mengorganisir, dan mengakses file. Hal tersebut kemudian mendiskusikan bagaimana file harus diperoleh dan bagaimana integritas dari file harus dipelihara. Bagian ini juga mendiskusikan berbagai isu teknis berhubungan dengan pemulihan dari suatu file, seperti pemulihan data dari file yang dihapus. Hal terakhir dari bagian ini menguraikan analisa dan pengambilan file, menyediakan bimbingan atas teknik dan *tools* yang dapat membantu analis.

## 4.1 File Dasar

Sebelum mencoba untuk memperoleh atau menguji file, analis sudah perlu memahami sedikitnya suatu dasar file dan *filesystems*. Bagian 4.1.2 menjelaskan bagaimana *filesystems* digunakan untuk mengorganisir file, dan menyediakan suatu ikhtisar beberapa *filesystems* umum. Bagian 4.1.3 mendiskusikan bagaimana data dari file yang dihapus masih terdapat di dalam *filesystems*. Analis perlu juga menyadari variasi media yang dapat berisi file; Bagian 4.1.1 menyediakan beberapa contoh media yang utama digunakan di dalam komputer pribadi dan contoh dari beberapa lebih media yang biasanya digunakan di dalam jenis alat digital lain.

### 4.1.1 Media penyimpanan File

Penggunaan komputer yang tersebar luas dan alat digital lain telah mengakibatkan suatu peningkatan penting di dalam banyaknya jenis media berbeda yang digunakan untuk menyimpan file. Sebagai tambahan terhadap jenis media yang biasa digunakan seperti *diskette* dan *hard drives*, file adalah suatu yang disimpan pada alat yang digunakan konsumen seperti

PDA dan telepon selular, seperti halnya jenis media yang lebih baru, seperti *flash card* yang dipopulerkan dengan adanya kamera digital. Tabel 4-1 mendaftarkan jenis media tersebut yang biasanya digunakan sekarang ini pada komputer dan alat digital. Daftar ini tidak meliputi tiap-tiap jenis media yang tersedia; melainkan, hal tersebut dimaksudkan untuk menunjukkan variasi jenis media yang mungkin seorang analis mengetahuinya.

Tabel 4.1. Tipe Media Yang Biasa Digunakan

| Tipe Media | Alat Pembaca | Kapasitas | Komentar |
|---|---|---|---|
| **Yang terutama digunakan dalam komputer pribadi** | | | |
| Floppy disk | Floppy disk drive | 1.44 megabytes (MB) | Disk berukuran 3.5 inchi; popularitasnya mulai menurun |
| CD-ROM | CD-ROM drive | 650 MB – 800 MB | Meliputi CD-R(sekali tulis) dan CD-RW (yang isinya dapat ditulisi kembali); media yang biasanya banyak digunakan |
| DVD-ROM | DVD-ROM drive | 1.67 gigabytes (GB) – 15.9 GB | Meliputi DVD-R(sekali tulis) dan DVD-RW (yang isinya dapat ditulisi kembali baik *single* dan *dual layer disks* |
| Hard drive | N/A | 20 GB – 300 GB | Driver dengan kapasitas sangat besar yang banyak digunakan dalam server file |
| Zip disk | Zip drive | 100 MB – 750 MB | Lebih besar daripada floppy disk |
| Jaz disk | Jaz drive | 1 GB – 2 GB | Serupa dengan Zip disks; tidak lagi diproduksi |
| Backup tape | Compatible tape drive | 80 MB – 320 GB | Banyak menyerupai kaset tape audio; sewajarnya peka terhadap kerusakan dalam kaitan dengan kondisi lingkungan |
| Magneto Optical (MO) disk | Compatible MO drive | 600 MB – 9.1 GB 5.25 | disk berukuran inci; lebih sedikit peka untuk kondisi lingkungan dibanding backup tape |
| ATA flash card | PCMCIA slot | 8 MB – 2 GB | PCMCIA flash memory card; berukuran 85.6 x 54 x 5 mm |
| **Digunakan oleh banyak jenis dari alat digital** | | | |
| Flash/Jump drive | USB interface | 16 MB – 2 GB | Also known as thumb drive because of their size |
| CompactFlash card | PCMCIA adapter atau memory card reader | 16 MB – 6 GB | Kartu tipe 1 berukuran 43 x 36 x 3.3 mm; kartu tipe 2 berukuran 43 x 36 x 5 mm |
| Microdrive r | PCMCIA adapter atau memory card reader | 340 MB – 4 GB | Alat penghubung dan bentuk faktor sama sebagai kartu Compactflash tipe 2 |
| MultiMediaCard (MMC) | PCMCIA adapter atau memory card reader | 16 MB – 512 MB | Berukuran 24 x 32 x 1.4 mm |
| Secure Digital (SD) Card | PCMCIA adapter atau memory card reader | 32 MB – 1 GB | Memenuhi kebutuhan dengan Secure Digital Music Initiative (SDMI); menyediakan data *built-in* isi file yang dienkripsi; dari luarnya serupa MMC |
| Memory Stick PCMCIA adapter or | memory card reader | 16 MB – 2 GB | Mencakup Memory Stick (50 x 21.5 x 2.8 mm), Memory Stick Duo (31 x 20 x 1.6 mm), MemoryStick PRO, Memory Stick PRO Duo; beberapa memenuhi kebutuhan SDMI dan menyediakan enkripsi *built-in* dari isi file |
| SmartMedia Card | PCMCIA adapter atau memory card reader | 8 MB – 128 MB | Berukuran 37 x 45 x 0.76 mm |
| xD-Picture Card | PCMCIA adapter atau xD- | 16 MB – 512 MB | Sekarang ini hanya digunakan di dalam kamera digital Fujifilm dan Olympus |

| Tipe Media | Alat Pembaca | Kapasitas | Komentar |
|---|---|---|---|
| | Picture card reader | | ; berukuran 20 x 25 x 1.7 mm |

### 4.1.2 Sistem File

Sebelum media dapat digunakan untuk menyimpan, biasanya media tersebut harus dipartisi dan diformat kedalam *logical volumes*. Mempartisi adalah suatu perbuatan yang sesuai untuk membagi suatu media kedalam ukuran – ukuran yang berfungsi sebagai fisik dari unit yang terpisah. *Logical volume* adalah partisi atau sebuah kumpulan dari proses partisi sebagai satu kesatuan yang telah diformat dengan suatu sistem file. Beberapa jenis media, seperti disket, dapat berisi paling banyak satu partisi (dan sebagai konsekuensi, satu *logical volumes*). Format dari *logical volumes* ditentukan oleh sistem file yang terpilih.

Suatu sistem file menggambarkan cara file dinamai, disimpan, diorganisir dan diakses pada *logical volumes*. Banyak perbedaan sistem file yang ada, masing-masing menyediakan fitur dan struktur data. Bagaimanapun, semua sistem file berbagi beberapa ciri umum. Pertama, mereka menggunakan konsep direktori dan file untuk mengorganisir dan menyimpan data. Direktori adalah struktur organ yang digunakan untuk menggolongkan file bersama-sama. Sebagai tambahan terhadap membuat file, direktori boleh berisi direktori lain yang disebut subdirektori. Kedua, sistem file menggunakan beberapa struktur data untuk menunjuk penempatan file pada media. Juga, mereka menyimpan masing-masing file data yang ditulis ke media dalam satu atau lebih unit alokasi file. Ini dikenal sebagai *cluster*s oleh beberapa sistem file (seperti, *File AllocationTable* [ FAT], *NT File System* [ NTFS]) dan blok oleh sistem file lainnya (seperti sistem file Unix dan Linux). Suatu unit alokasi file adalah suatu kelompok yang sederhana dari sektor, yang merupakan unit yang paling kecil yang dapat diakses pada suatu suatu media.

Materi berikut menguraikan beberapa hal yang digunakan filesystems:

1. FAT12.3 FAT12 digunakan hanya pada disket dan volume FAT yang lebih kecil dibanding 16 MB. FAT12 menggunakan suatu 12-bit file alokasi *table entry* untuk menunjuk suatu isi sistem file itu.
2. FAT16. MS DOS, Windows 95/98/Nt/2000/Xp, Server Windows 2003, dan beberapa sistem operasi UNIX mendukung FAT16 secara asli. FAT16 biasanya juga digunakan untuk alat multimedia seperti *audio player* dan kamera digital. FAT16 menggunakan suatu 16-bit file alokasi *table entry* untuk menunjuk suatu isi sistem file itu. Volume FAT16 terbatas pada suatu ukuran maksimum 2 GB di dalam MS DOS dan Windows

95/98. Windows NT dan sistem operasi yang lebih baru meningkatkan ukuran volume yang maksimum untuk FAT16 menjadi 4 GB.

3. FAT32.4 Windows 95 OEM Service Release 2(OSR2), Windows 98/2000/Xp, dan Server Windows 2003 yang mendukung FAT32 secara asli, seperti halnya beberapa alat multimedia. FAT32 menggunakan suatu 32-bit file alokasi *table entry* untuk menunjuk suatu isi sistem file itu. Ukuran maksimum volume FAT32 adalah 2 terabytes ( TB).
4. NTFS. Windows NT/2000/XP dan Server Windows 2003 mendukung NTFS secara asli. NTFS adalah suatu sistem file dapat dipulihkan, yang berarti bahwa hal tersebut dapat secara otomatis mengembalikan konsistensi dari sistem file manakala terjadi kesalahan. Sebagai tambahan, NTFS mendukung data yang dikompres dan telah dienkripsi dan mengijinkan user dan level grup mengakses ijin untuk digambarkannya file data dan direktori - direktori. Ukuran maksimum volume NTFS adalah 2 TB.
5. High-Performance File System ( HPFS). HPFS didukung secara langsung oleh OS/2 dan dapat dibaca oleh Windows NT 3.1, 3.5, dan 3.51. HPFS membangun atas organisasi direktori tentang FAT dengan menyediakan penyortiran direktori secara otomatis. Sebagai tambahan, HPFS mengurangi jumlah ruang disk yang hilang dengan pemanfaatan unit alokasi lebih kecil. Ukuran volume maksimum HPFS adalah 64 GB.
6. Second Extended Filesystem ( ext2fs). ext2fs didukung secara langsung oleh Linux. ext2fs mendukung jenis file standar Unix dan cek sistem file untuk memastikan konsistensi dari sistem file. Ukuran volume maksimum ext2fs adalah 4 TB.
7. Third Extended Filesystem ( ext3fs). ext3fs didukung secara langsung oleh Linux. ext3fs didasarkan pada ext2fs filesystem dan menyediakan kemampuan menjurnal yang mengijinkan cek konsistensi menyangkut sistem file untuk dilakukan dengan cepat pada sejumlah data yang besar. Ukuran volume maksimum ext3fs adalah 4 TB.
8. Hierarchical File System ( HFS). HFS didukung secara langsung oleh Mac OS. HFS sebagian besar digunakan di dalam versi Mac OS yang terdahulu tetapi masih didukung dalam versi lebih baru. Ukuran volume maksimum HFS di bawah Mac OS 6 dan 7 adalah 2 GB. Ukuran volume maksimum HFS dalam Mac OS 7.5 adalah 4 GB. Mac O 7.5.2 dan sistem operasi Mac yang terbaru meningkatkan ukuran volume maksimum HFS menjadi 2 TB.

9. HFS Plus.8 HFS lebih didukung secara asli oleh Mac OS 8.1 dan versi selanjutnya dan hal tersebut merupakan suatu jurnal sistem file di bawah Mac OS X. HFS Plus adalah pengganti HFS dan menyediakan banyak peningkatan seperti mendukung nama file yang panjang dan nama file Unicode mendukung untuk penamaan file internasional. Ukuran volume maksimum HFS Plus adalah 2 TB.
10. Unix File System ( UFS). UFS didukung secara asli oleh beberapa jenis sistem operasi Unix, termasuk Solaris, FreeBSD, OpenBSD, dan Mac OS X. Bagaimanapun, kebanyakan sistem operasi sudah menambahkan fitur kepemilikan, sehingga detil UFS berbeda antar implementasi.
11. Compact Disk File System ( CDFS). Seperti indikasi dari namanya, CDFS sistem file digunakan untuk CD.
12. International Organization for Standardization ( ISO) 9660. ISO 9660 sistem file biasanya digunakan pada CD-ROMS. Sistem file CD-ROM yang popular lainnya adalah Joliet, suatu varian ISO9660. ISO 9660 mendukung panjangnya nama file sampai 32 karakter, selagi Joliet mendukung sampai kepada 64 karakter. Joliet juga mendukung karakter Unicode di dalam pemberian nama file.
13. Universal Disk Format ( UDF). UDF adalah sistem file yang digunakan untuk DVD dan juga digunakan untuk beberapa CD.

### 4.1.3 Data lain pada Media

Seperti yang diuraikan di dalam Bagian 4.1.2, sistem file dirancang untuk menyimpan file pada suatu media. Bagaimanapun, sistem file dapat juga menjaga data dari dihapusnya file atau versi yang lebih awal dari pengadaan file. Data ini dapat menyediakan informasi penting. (Bagian 4.2 mendiskusikan teknik untuk memperoleh data). Materi berikut menguraikan bagaimana data ini masih dapat tersisa pada suatu media:

1. File yang dihapus. Manakala suatu file dihapus, hal tersebut secara khusus tidak dihapus dari media; sebagai gantinya, informasi di dalam direktori - direktori struktur data yang menunjuk kepada penempatan dari file ditandai ketika dihapus. Ini berarti file masih disimpan pada media tetapi hal itu tidak lagi dihitung oleh sistem operasi tersebut. Bagaimanapun, sistem operasi mempertimbangkan ini untuk menjadi ruang kosong dan dapat di-*overwrite* bagian manapun menyangkut file yang dihapus pada setiap waktu.
2. *Slack Space*. Seperti dicatat sebelumnya, sistem file menggunakan unit – unit alokasi file untuk menyimpan file. Sekalipun suatu file memerlukan lebih sedikit ruang dibanding

ukuran unit alokasi file, secara keseluruhan unit alokasi file masih disediakan untuk file itu. Sebagai contoh, jika ukuran unit alokasi file adalah 32 KB dan suatu file hanya 7 KB, keseluruhan 32 KB masih dialokasikan untuk file tetapi hanya 7 KB digunakan, menghasilkan 25 KB ruang yang tak terpakai. Ruang yang tak terpakai ini dikenal sebagai file Slack Space dan mungkin menjaga data yang bersifat sisa seperti bagian dari file yang dihapus.

3. *Free Space*. Free Space atau ruang kosong adalah area pada media yang tidak dialokasikan untuk membagi apapun dan terdiri dari *clusters* yang tidak teralokasi atau blok. Hal ini sering meliputi ruang pada media dimana file (dan bahkan keseluruhan volume) mungkin telah diletakkan dengan satu petunjuk tetapi sejak itu filenya telah dihapus. Ruang kosong masih dapat berisi potongan data.

Cara yang lain dimana data dapat disembunyikan adalah dengan Alternate Data Streams ( ADS ) di dalam volume NTFS. NTFS telah lama didukung berbagai arus data untuk direktori dan file. Masing-Masing file pada suatu volume NTFS terdiri dari dari suatu *stream* tak dikenal yang digunakan untuk menyimpan data primer file dan secara bebas pilih satu atau lebih menamai *stream* ( yaitu, file.txt:Stream1, file.txt:Stream2) yang dapat digunakan untuk menyimpan alat bantu informasi seperti properti dari file dan gambar *thumbnail* dari sebuah data. Sebagai contoh, jika seorang *userright-clicks* pada suatu file di dalam Windows Explorer, melihat properti daripada file dan kemudian memodifikasi informasi yang diperlihatkan di dalam *summary tab*, OS menyimpan ringkasan informasi untuk file di dalam suatu arus yang dinamai.

Semua arus data di dalam suatu file berbagi atribut file ( seperti, *timestamps,* atribut keamanan). Walaupun nama *stream* mempengaruhi bagian penyimpanan dari suatu file, mereka sebagian besar dirahasiakan dari para *user* sebab *standard Windows file utilities* seperti *Explorer* hanya melaporkan ukuran dari suatu *stream* file tak dikenal.Oleh karena itu, seorang user tidak bisa siap menentukan jika suatu file berisi ADS menggunakan *standard Windows file utilities*. Hal ini mengijinkan data yang tersembunyi untuk dimasukkan ke dalam sistem file NTFS manapun. Menggerakkan file dengan ADS ke sistem file non-NTFS secara efektif melepaskan ADS dari file, maka ADS dapat hilang jika analis tidak peduli akan kehadiran mereka. Perangkat lunak dan proses tersedia untuk mengidentifikasi ADS.

## 4.2 Memperoleh File

Selama mendapatkan data, analis perlu membuat satu atau lebih salinan menyangkut file yang diinginkan atau sistem file. Analis kemudian bisa bekerja dengan suatu salinan file tanpa mempengaruhi file yang asli. Bagian 4.2.1 menguraikan *tools* dan teknik yang utama untuk penyalinan file dan data file yang merupakan sisa dari suatu media. Bagian 4.2.2 mendiskusikan pentingnya pemeliharaan integritas dari memfile dan menyediakan bimbingan atas perangkat keras dan lunak yang dapat membantu dengan memelihara dan membuktikan integritas file. Hal itu sering penting untuk memperoleh sesuatu tidak hanya file, tetapi *timestamps* juga penting untuk file, seperti ketika file sedang diakses atau pada akhir dimodifikasi; Bagian 4.2.3 menguraikan *timestamps* dan menjelaskan bagaimana mereka dapat dipelihara. Isu teknik lain yang berhubungan dengan mendapatkan suatu data, seperti temuan file yang tersembunyi dan penyalinan file - file dari implementasi Redundant Arrays of Inexpensive Disks ( RAID), ditunjukan dalam Bagian 4.2.4.

### 4.2.1 Penyalinan file dari media

File dapat disalin dari media yang menggunakan dua teknik berbeda, sebagai berikut:

1. *Logical Backup*. Suatu *logical Backup* menyalin file dan direktori dari suatu *logical volumes*. Hal tersebut tidak menangkap data lain yang mungkin disajikan pada media, seperti file yang dihapus atau data sisa yang disimpan di dalam *slack space.*
2. *Physical Backup*. Juga yang dikenal sebagai *disk imaging*, suatu *physical backup* yang menghasilkan suatu salinan bit-for-bit media yang asli, mencakup *free space* dan *slack space*. *Physical Backup* memerlukan lebih ruang penyimpanan dan mengambil waktu lebih panjang untuk pelaksanaannya dibanding *logical backups*.

Ketika suatu *physical backup* dieksekusi, baik suatu *disk-to-disk* maupun suatu *disk-to-file copy* dapat dilakukan. Suatu *disk-to-disk copy*, sebab hal tersebut mengesankan namanya, menyalin muatan dari media secara langsung ke media yang lain . Suatu *disk-to-file copy* menyalin muatan dari media ke file data tunggal yang logis. Suatu *disk-to-disk copy* bermanfaat sejak media yang disalin dapat dihubungkan secara langsung ke suatu komputer dan muatannya siap untuk dilihat. Bagaimanapun, suatu *disk-to-disk copy* memerlukan suatu media kedua yang serupa kepada media yang asli. Suatu *disk-to-file copy* mengijinkan gambaran file data untuk dipindahkan dan di-*backup* dengan mudah. Bagaimanapun, untuk melihat muatan yang logis dari suatu file gambaran, analis harus mengembalikan gambaran ke

media atau membukanya di dalam suatu aplikasi yang mampu memperlihatkan muatan logis yang menggambarkan suatu file data. Bagian 4.3 mendiskusikan hal ini secara lebih detil.

Banyak *tools* perangkat keras dan lunak yang dapat melaksanakan *physical* dan *logical backups*. *Tools* perangkat keras biasanya bersifat *portable*, menyediakan *bit-by-bit* gambaran, menghubungkan secara langsung kepada komputer atau *drive* untuk digambarkan, dan mempunyai banyak fungsi *built-in hash*. *Tools* perangkat keras dapat memperoleh data dari *drive* yang menggunakan pengontrol jenis umum, seperti *Integrated Drive Electronics* ( IDE) dan *Small Computer System Interface* ( SCSI). Solusi perangkat lunak yang biasanya terdiri dari suatu disket *startup*, CD atau menginstall program yang dijalankan pada *workstation* untuk media mana yang diberikan untuk dapat diberi gambaran. Beberapa solusi perangkat lunak menciptakan salinan file atau bagian - bagian yang logis dan dapat mengabaikan ruang *drive* yang tidak teralokasi atau cuma-cuma, ketika yang lain menciptakan suatu *bit-by-bit* gambaran salinan media tersebut. Jenis data yang diperlukan dapat menentukan *tools* perangkat lunak atau perangkat keras yang digunakan untuk *data imaging*. Sebagai contoh, seandainya satu *folder* pada bagian tertentu diperlukan, analis bisa menggunakan suatu solusi perangkat lunak sederhana sebagai ganti suatu alat *hardware-based imaging*.

Dalam kaitan dengan tersedianya sejumlah *disk imaging tools* yang terus meningkat dan ketiadaan suatu standar untuk pengujiannya, proyek NIST Computer Forensics Tool Testing (CFTT) telah mengembangkan prosedur pengujian tepat untuk memvalidasi hasil dari *tools* tersebut. Ssekarang ini, hanya sedikit *disk imaging tools* yang sudah mengalami tes CFTT. Beberapa *disk imaging tools* dapat juga melaksanakan *recordkeeping*, seperti jejak audit yang diaotomatiskan . Penggunaan *tools* seperti itu dapat mendukung konsistensi di dalam proses pengujian dan ketelitian dan hasil dari pengembangan.

Biasanya, *tools* yang melaksanakan *physical backup's* harusnya tidak digunakan untuk memperoleh salinan *bit-by-bit* dari suatu keseluruhan *physical device* dari suatu tempat sistem—sistem yang sekarang ini digunakan—karena memori dan file pada sistem yang demikian sedang berubah secara konstan dan oleh karena itu tidak bisa divalidasi. Bagaimanapun, suatu salinan *bit-by-bit* dari area yang logis dari tempat suatu sistem dapat diselesaikan dan divalidasi. Manakala *logical backups* sedang dilakukan, hal tersebut masih lebih baik untuk tidak menyalin file dari tempat suatu sistem; perubahan dapat terjadi pada file sepanjang proses *backup*, dan file yang dipegang terbuka oleh suatu proses yang mungkin tidak memudahkan untuk menyalin. Maka, analis perlu memutuskan apakah penyalinan file

dari tempat suatu sistem mungkin didasarkan pada file yang perlu untuk diperoleh, bagaimana penyalinan akurat dan lengkap diperlukan, dan seberapa penting sistem tempat. [5]Sebagai contoh, hal itu tidak diperlukan untuk melepas suatu *critical server* yang digunakan oleh ratusan orang hanya untuk memperoleh file dari *single user's home directory*. Untuk *logical backups* dari *live systems*, analis dapat menggunakan standar sistem *backup* perangkat lunak. Bagaimanapun, melakukan suatu *backup* bisa berdampak pada pencapaian dari sistem dan menghabiskan sejumlah *network bandwidth* yang penting, tergantung pada apakah backup dilakukan secara local atau secara jarak jauh.

Organisasi perlu mempunyai kebijakan dan prosedur yang menandai adanya keadaan di bawah *physical* dan *logical backups* ( termasuk *live systems*nya) yang mungkin dilakukan dan personil mana yang boleh melaksanakan hal tersebut. Hal tersebut secara khusus efektif untuk menetapkan kebijakan dan prosedur berdasar pada kategori sistem (yaitu, rendah, medium, atau dampak yang tinggi) dan sifat alami suatu peristiwa yang menarik; beberapa organisasi mungkin juga memilih untuk menciptakan statemen kebijakan terpisah dan prosedur untuk sistem yang penting. Prosedur atau kebijakan perlu mengidentifikasi individu atau kelompok dengan otoritas untuk membuat keputusan mengenai *backups*; orang ini harus mampu untuk menimbang resiko dan membuat keputusan penting. Prosedur atau kebijakan perlu juga mengidentifikasi kelompok atau individu yang mempunyai otoritas untuk melaksanakan *backup* untuk masing-masing jenis sistem; mengakses ke beberapa sistem boleh jadi terbatas oleh karena kepekaan dari data atau operasi di dalam sistem itu.

### 4.2.2 Integritas File Data

Selama *backups*, integritas dari keaslian media harus dipelihara. Untuk memastikan bahwa proses backups tidak merubah data pada media yang asli, analis dapat menggunakan *write-blocker* ketika mem*back-up* media. *Write-blocker* adalah *tools* dasar perangkat keras atau perangkat lunak yang mencegah komputer dari kegiatan menulis ke media penyimpanan computer yang terhubung dengannya. Perangkat keras *write-blockers* adalah suatu yang secara fisik terhubung dengan komputer dan media penyimpanan yang sedang diproses untuk mencegah penulisan apapun ke media tersebut. Perangkat lunak *write-blockers* adalah suatu yang diinstall pada system analisa dan yang sekarang tersedia hanya untuk MS-DOS dan sistem Windows. ( Banyak system operasi ( misalnya Mac OS X) yang mungkin tidak memerlukan perangkat lunak *write-blockers* karena mereka telah di set untuk proses boot dengan alat sekunder yang tidak tersusun. Bagaimanapun, memasang suatu alat perangkat

keras *writeblocking* akan memastikan integritas yang dirawat.) Perangkat lunak MS-DOS-based write-blockers bekerja dengan pedoman trapping Interrupt dan memperluas Interrupt disk writes. Perangkat lunak Windows-based write-blockers menggunakan filter untuk mengurutkan jenis Interrupt yang dikirim ke alat untuk mencegah tulisan apapun ke media penyimpanan.

Secara umum, manakala penggunaan suatu perangkat keras *write-blocker*, alat atau penggunaan media untuk dibaca media seharusnya dihubungkan secara langsung kepada *write-blocker*, dan *write-blocker* harus dihubungkan kepada komputer atau alat yang digunakan untuk melaksanakan backup itu. Manakala penggunaan suatu perangkat lunak *write-blocker*, perangkat lunak harus sudah ada dalam komputer sebelum alat atau media yang digunakan untuk membaca media dihubungkan kepada komputer itu. *Writeblockers* boleh juga mengijinkan *write-blocking* untuk menjadi *toggled on* atau *off* untuk alat tertentu . Hal itu penting ketika write-blocking digunakan, yang berarti *toggled on* untuk semua alat yang dihubungkan. *Write-Blockers* seharusnya diuji secara rutin untuk memastikan mereka mendukung alat lebih baru. Sebagai contoh, suatu alat baru mungkin menggunakan fungsi sebelumnya tak terpakai atau yang dipesan atau *placeholders* untuk menerapkan fungsi alat yang spesifik yang akhirnya dapat menulis ke suatu alat dan mengubah muatannya.

Setelah suatu *backup* dilakukan, maka penting untuk mem buktikan bahwa data yang di*copy* adalah suatu salinan yang tepat menyangkut data yang asli. Menghitung intisari pesan dari data di*copy* dapat digunakan untuk membuktikan dan memastikan integritas data. Suatu intisari pesan adalah suatu tanda digital yang dengan uniknya mengidentifikasi data dan mempunyai hak milik yang mengubah bit tunggal di dalam data yang akan menyebabkan dihasilkannya intisari suatu pesan yang berbeda. Ada banyak algoritma untuk menghitung intisari pesan data, tetapi ada dua paling umum digunakan adalah MD5 dan Secure Hash Algorithm ( SHA-1). Algoritma ini diambil sebagai data input *arbitrary length* dan hasil seperti keluaran 128-bit intisari suatu pesan. Sebab SHA-1 adalah suatu algoritma FIPS-APPROVED dan bukan MD5, Para agen pemerintah pusat perlu menggunakan SHA-1 sebagai ganti MD5 untuk intisari pesan .

Manakala suatu *physical backup* dilakukan, intisari suatu pesan dari media asli harus dihitung dan direkam sebelum *backup* dilakukan. Setelah *backup* dilakukan, intisari suatu pesan dari media di*copy* harus dihitung dan dibandingkan dengan intisari pesan yang asli untuk membuktikan integritas data itu telah dipelihara. Apalagi, intisari suatu pesan dari

media asli harus dihitung lagi untuk membuktikan bahwa proses *backup* tidak mengubah media yang asli, dan semua hasil harus didokumentasikan. Proses harus digunakan untuk *logical backups*, kalau tidak intisari pesan harus dihitung dan dibandingkan untuk masing-masing file data.

### 4.2.4 Modifikasi File, Akses, dan Waktu penciptaan

Hal tersebut sering menjadi penting untuk mengetahui kapan suatu file digunakan atau dimanipulasi dan kebanyakan sistem operasi menjejaki *timestamps* tertentu yang berhubungan dengan file. *Timestamps* yang biasanya digunakan adalah modifikasi, akses, dan waktu penciptaan (modification, access,
and creation; MAC), sebagai berikut:

1. Waktu Modifikasi. Ini adalah waktu yang terakhir suatu file diubah dengan berbagai cara. Hal ini meliputi ketika suatu file dituliskan dan ketika file tersebut diubah oleh program yang lain .
2. Akses Waktu. Ini adalah waktu yang terakhir untuk dilakukannya akses apapun pada suatu file ( misalnya, dilihat, dibuka, dicetak).
3. Waktu penciptaan. Hal ini biasanya merupakan waktu dan tanggal file ketika diciptakan. Bagaimanapun, manakala suatu file di*copy* untuk suatu sistem, Waktu penciptaan akan menjadi waktu dari file yang di*copy* kepada sistem yang baru itu.Waktu modifikasi akan tetap utuh.

Masing-Masing jenis *filesystem* boleh menyimpan jenis waktu yang berbeda. Sebagai contoh, Sistem *Windows* mempertahankan waktu yang dimodifikasi terakhir, waktu akses terakhir, dan waktu penciptaan file. Sistem UNIX mempertahankan modifikasi terakhir, perubahan *inode* terakhir, dan waktu akses terakhir. Bagaimanapun, beberapa sistem UNIX ( termasuk versi BSD dan Sunos) tidak memperbaharui waktu akses terakhir dari file yang dapat dijalankan manakala di-*run*. Beberapa sistem UNIX merekam waktu manakala metadata untuk suatu file diubah menjadi paling akhir. Metadata adalah data yang menyediakan informasi tentang suatu muatan file.

Jika suatu analis ingin menetapkan timeline yang akurat dari suatu peristiwa, maka file harus dipelihara untuk memperoleh hasil yang dapat dipercaya. Analis harus sadar bahwa tidak semua metode untuk memperoleh file data dapat memelihara waktu dari file. Sebagai contoh, melakukan suatu *logical backups* dapat menyebabkan waktu diciptakannya file

menjadi diubah manakala suatu file data di*copy*, tetapi *physical backup* dapat memelihara waktu dari file sebab suatu *bit-for-bit copy* telah dihasilkan.Oleh karena itu, kapan saja waktu dari file penting, suatu *physical backup* harus digunakan untuk memperoleh data. Analis harus sadar waktu dari file itu tidak mungkin akurat untuk berbagai pertimbangan, mencakup hal berikut:

1. Jam komputer tidak mempunyai waktu yang benar itu. Sebagai contoh, jam tidak mungkin sama secara teratur dengan suatu sumber waktu yang memiliki kewenangan.
2. Waktu tidak mungkin direkam dengan tingkat detil yang diharapkan, seperti itu dapat menghilangkan detik atau beberapa menit.
3. Suatu *attacker* mungkin telah mengubah waktu file yang direkam.

### 4.2.4 Isu Teknis

Ada beberapa isu teknis yang dapat muncul dalam memperoleh file data. Seperti yang diuraikan dalam Bagian 4.2.1, isu utama adalah memperoleh sisa-sisa file yang masih ada dan file yang dihapus dalam *free space* dan *slack space* pada suatu media. Individu dapat menggunakan berbagai teknik untuk merintangi pengadaan data seperti itu. Sebagai contoh, ada banyak kegunaan yang tersedia dalam melaksanakan *wiping—overwriting* media ( atau bagian dari media, seperti file tertentu) dengan nilai-nilai tetap atau acak ( misalnya, semua 0). Kegunaan seperti itu bertukar-tukar dalam reliabilitas dan keandalan, tetapi yang paling efektif adalah di dalam mencegah perolehan yang mudah atas suatu file, yang terutama jika beberapa *wipes* yang dilakukan. Individu dapat juga menggunakan alat-alat fisik untuk mencegah didapatnya data, seperti *demagnetizing* suatu *harddrive* ( juga yang dikenal sebagai *degaussing*) atau secara fisik merusakkan atau membinasakan media. Kedua fisik dan teknik *software-based* dapat membuatnya sangat sulit, atau bahkan mustahil, untuk memulihkan semua data yang menggunakan perangkat lunak. Mencoba lakukan pemulihan dalam kasus ini mengharuskan penggunaan dari tenaga khusus ahli forensik dengan fasilitas yang telah maju, perangkat keras, dan teknik, tetapi usaha dan biaya dalam pelaksanaannya menjadi penghalang untuk menggunakan cara ini secara keseluruhan.[10] Dalam beberapa hal, data sederhananya tidak dapat dipulihkan.

Isu umum lainnya adalah memperoleh data yang tersembunyi. Banyak sistem operasi mengijinkan para user ke label file tertentu, direktori, atau bagian tetap yang tersembunyi, yang mana dengan kesalahan, mereka tidak ditampilkan di dalam listing direktori. [11]Beberapa sistem operasi dan aplikasi menyembunyikan konfigurasi dari file untuk mengurangi

kesempatan para user yang akan secara kebetulan memodifikasi atau menghapusnya. Juga, pada beberapa sistem operasi, direktori yang telah dihapus mungkin telah ditandai secara tersembunyi. Data yang tersembunyi dapat berisi banyak informasi; sebagai contoh, suatu partisi yang tersembunyi bisa berisi suatu sistem operasi terpisah dan banyak file data. [12]Para user dapat menciptakan partisi yang tersembunyi dengan mengubah tabel partisi untuk mengganggu manajemen disk dan mencegah aplikasi dari melihat adanya area data. Data yang tersembunyi dapat juga ditemukan di dalam ADSs pada volume NTFS dan *end-of-file slack space* dan *free space* pada suatu medium. Banyak *tool* untuk mendapatkan data yang dapat mengenali beberapa atau semua metode ini dalam menyembunyikan data dan memulihkan data yang terhubung.

Isu lain yang masih dapat muncul adalah yang memperoleh data dari array RAID yang menggunakan *striping* ( e.g., RAID-0, RAID-5). Di dalam konfigurasi ini, suatu volume yang terbagi terdiri dari bagian *equal-sized* yang berada pada *disk drive* terpisah. Kapan data ditulis ke volume, hal itu biasanya partisi didistribusi silang untuk meningkatkan performa disk. Hal ini akan menjadi suatu masalah sebab semua partisi dari suatu volume terbagi yang harus diahdirkan untuk menguji muatannya, tetapi bagian yang berada pada *physical disk drives* terpisah. Oleh karena itu, untuk menguji suatu volume yang terbagi, masing-masing disk drive di dalam array RAID perlu untuk digambarkan dan konfigurasi RAID dikonfigurasi kembali pada pada pengujian sistem. Beberapa tool dapat memperoleh volume yang dibagi – bagi dan tool juga mampu memelihara area data tak terpakai dari sebuah volume, seperti free space dan slack space .

## 4.3 Pengujian File Data

Setelah *logical* atau *physical backup* dilakukan, *backup* mungkin telah dikembalikan ke media lain sebelum data dapat diuji. Hal ini bergantung pada *tools* yang akan digunakan untuk melaksanakan analisa itu. Beberapa *tool* dapat meneliti data secara langsung dari suatu file gambaran, ketika yang lain memerlukan gambaran *backup* dikembalikan medium yang dulu. Dengan mengabaikan apakah suatu *backup* gambaran file atau dikembalikannya gambaran yang digunakan di dalam pengujian, hal tersebut hanya perlu diakses sebagai *read-only* untuk memastikan bahwa data yang sedang diuji tidak dimodifikasi dan data akan menyediakan hasil konsisten pada urutan yang dijalankan. Seperti dicatat dalam bagian 4.2.2, *write-blockers* dapat digunakan selama proses ini untuk mencegah dari terjadinya penulisan

kepada pengembalian gambaran tersebtu. Setelah pengembalian backup ( jika diperlukan), analis mulai untuk menguji data yang diperoleh dan melaksanakan suatu penilaian tentang data dan file yang relevan dengan penempatan semua file, termasuk menghapus file, sisa-sisa file dalam *slack* dan *free space*, dan file yang disembunyikan. Berikutnya, analis boleh harus mengakses data di dalam file, yang mana mungkin sampai hal yang diperumit seperti langkah mengenkripsi dan kata sandi. Bagian ini menguraikan proses seperti halnya teknik yang dapat mempercepat pengujian data dan file.

### 4.3.1 Menempatkan File

Langkah pertama dalam pengujian adalah menempatkan file. Suatu *disk image* mungkin menangkap banyak gigabytes *free space* dan *slack space*, yang mana bisa berisi ribuan file dan fragmen file. Yang secara manual mengekstrak data dari ruang yang tak terpakai bisa merupakan suatu proses sulit dan memakan waktu, hal seperti itu memerlukan pengetahuan yang mendasari format sistem file. Yang secara kebetulan, beberapa tool tersedia untuk dapat mengotomatiskan proses mengekstrak ruang yang tak terpakai dan menyimpannya ke file data, seperti halnya pemulihan file yang dihapus dan file di dalam *recycle bin*. Analis dapat juga memperlihatkan *slack space* dengan *hex editors* atau *special slack recovery tools.*

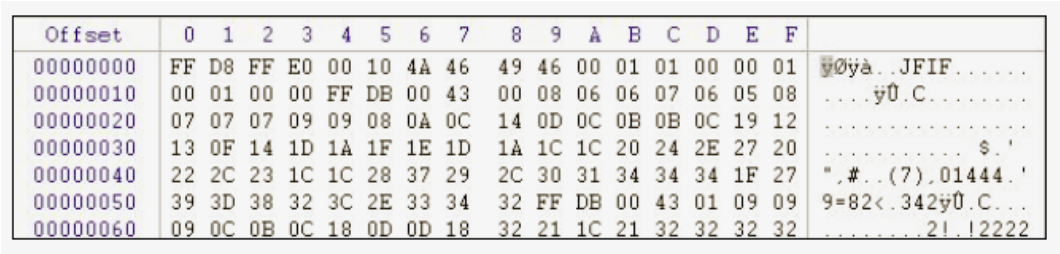
### 4.3.2 Mengakses Data

Langkah berikutnya dalam proses pengujian adalah memperoleh akses ke data di dalam file itu. Untuk bisa dipertimbangkan isi dari suatu file, suatu analis harus mengetahui seperti apa data yang merupakan isi dari file. Tujuan yang diharapkan dari ekstensi file adalah untuk menandakan sifat alami isi file; sebagai contoh, suatu perluasan jpg menandai adanya suatu file grafis, dan suatu perluasan mp3 menandai adanya suatu file musik. Bagaimanapun, para userdapat memberikan ekstensi atau perluasan untuk macam – macam jenis file, seperti penamaan suatu teks memfile mysong.mp3 atau penghilangan suatu ekstensi file. Juga, beberapa ekstensi file boleh jadi disembunyikan atau tidak mendukung pada sistem operasi lain. Oleh karena itu, analis mestinya tidak berasumsi bahwa ekstensi file akurat.

Analis dapat dengan teliti mengidentifikasi jenis data disimpan di dalam banyak file dengan memperhatikan *file header*nya. Suatu *file header* berisi tentang mengidentifikasi informasi tentang suatu file dan mungkin metadata yang menyediakan informasi tentang isi file. Seperti yang ditunjukkan di dalam Gambar 4-1, file header file berisi suatu tanda file

yang mengidentifikasi jenis data yang yang merupakan bagian dari isi file. *File header* bersifat menandakan untuk isi file dengan mengabaikan ekstensi dari file. Contoh di dalam Gambar 4-1 mempunyai *file header* dari FF D8, yang mana menunjukkan bahwa ini adalah suatu file JPEG. *File header* file bisa ditempatkan di dalam suatu file terpisah dari data file yang nyata.

Teknik efektif lain untuk mengidentifikasi jenis data di dalam suatu file adalah suatu histogram sederhana yang mempertunjukkan distribusi nilai ASCII sebagai persen dari total karakter di dalam suatu file. Sebagai contoh, suatu simbol di dalam ‘ space’, ‘ a’, dan ‘ yang merupakan bentuk biasanya menandai adanya suatu file teks, ketika konsistensi silang histogram menandai adanya suatu file yag dikompres. Pola baik lainnya untuk menandakan file yang dienkripsi atau sampai yang dimodifikasi adalah *steganography.*



| Offset | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00000000 | FF | D8 | FF | E0 | 00 | 10 | 4A | 46 | 49 | 46 | 00 | 01 | 01 | 00 | 00 | 01 | ÿØÿà..JFIF..... |
| 00000010 | 00 | 01 | 00 | 00 | FF | DB | 00 | 43 | 00 | 08 | 06 | 06 | 07 | 06 | 05 | 08 | ....ÿÛ.C..... |
| 00000020 | 07 | 07 | 07 | 09 | 09 | 08 | 0A | 0C | 14 | 0D | 0C | 0B | 0B | 0C | 19 | 12 | ..... |
| 00000030 | 13 | 0F | 14 | 1D | 1A | 1F | 1E | 1D | 1A | 1C | 1C | 20 | 24 | 2E | 27 | 20 | ..... $.' |
| 00000040 | 22 | 2C | 23 | 1C | 1C | 28 | 37 | 29 | 2C | 30 | 31 | 34 | 34 | 34 | 1F | 27 | ",#..(7),01444.' |
| 00000050 | 39 | 3D | 38 | 32 | 3C | 2E | 33 | 34 | 32 | FF | DB | 00 | 43 | 01 | 09 | 09 | 9=82<.342ÿÛ.C... |
| 00000060 | 09 | 0C | 0B | 0C | 18 | 0D | 0D | 18 | 32 | 21 | 1C | 21 | 32 | 32 | 32 | 32 | ........2!.!2222 |

Gambar 4-1.Informasi File Header

*Encryption* sering menghadirkan tantangan untuk para analis. Para *user*mungkin mengenkripsi file sendiri, folders, volume, atau partisi yang sedemikian sehingga orang lain tidak bisa mengakses isinya tanpa suatu kunci *decryptio*n atau passphrase. *Encryption* mungkin dilakukan oleh sistem operasi atau program pihak ketiga. Walaupun itu secara relatif mudah untuk mengidentifikasi suatu file yang dienkripsi, hal itu pada umumnya tidak mudah untuk mendekripsikannya. Analis mungkin mampu mengidentifikasi metode *encryption* dengan pengujian file header, mengidentifikasi program *encryption* yang diinstal pada suatu sistem, atau menemukan kunci enkripsi ( yang mana  sering disimpan pada media lain). Sekalinya metode enkripsi dikenal, analis dapat lebih baik menentukan kelayakan dari pendeskripsian file itu.

Walaupun dengan mudah suatu analis dapat mendeteksi kehadiran dari data yang dienkripsi, penggunaan *steganography* lebih sukar untuk mendeteksi. *Steganography*, juga dikenal sebagai *steg*, adalah menempelkan data di dalam data lain. *Digital watermarks* dan menyembunyikan kata-kata dan informasi dengan gambaran adalah contoh steganography. Beberapa teknik yang mungkin digunakan analis untuk menempatkan *data stegged* meliputi

pencarian berbagai versi gambaran yang sama, mengidentifikasi kehadiran gambaran grayscale, mencari dan mencatat metadata, menggunakan histogram, dan menggunakan *hash* yang ditetapkan untuk mencari-cari perangkat lunak *steganography yang* dikenal. Sekalinya terdapat data *stegged*, analis dapat menjadi mampu untuk mengekstrak data yang *embedded* dengan menentukan perangkat lunak apa yang menciptakan data dan kemudian menemukan kunci *stego*, atau penggunaan *brute force* dan serangan *cryptographic* untuk menentukan suatu *password*. Bagaimanapun, usaha ini sering gagal dan sangat memakan waktu, terutama sekali jika analis tidak temukan kehadiran dari perangkat lunak *steganography* yang dikenal pada media yang sedang ditinjau. Juga, beberapa program perangkat lunak dapat meneliti file dan menaksir kemungkinan bahwa file diubah dengan *steganography*.

Analis dapat juga berkeharusan mengakses file *non-stegged* yang dilindungi oleh kata sandi. Kata sandi sering disimpan pada sistem yang sama sebagai file yang mereka melindungi, tetapi di dalam suatu format dienkripsi atau disandikan. Berbagai kegunaan yang tersedia itu bisa juga terdapat kelemahan (yaitu *crack*) atas kata sandi yang ditempatkan pada file individu, seperti halnya *passwords* sistem operasi . Kegunaan meng*cracking* dapat dicoba untuk mengira kata sandi, seperti halnya melakukan usaha *brute force* yang mencoba tiap-tiap kata sandi mungkin. Waktu yang diperlukan untuk suatu serangan *brute force* pada suatu kata sandi yang dienkripsi atau disandikan dapat sangat bermacam – macam tergantung pada jenis enkripsi yang digunakan dan kesempurnaan dari kata sandinya sendiri. Pendekatan yang lain dalam beberapa peristiwa adalah mem*bypass* suatu kata sandi. Sebagai contoh, suatu analis dapat mem*boot* suatu sistem dan melumpuhkan kata sandi *screensaver*nya, atau mem*bypass* suatu kata sandi BIOS dengan penarikan BIOS jumper dari sistem *motherboard* atau penggunaan penghasil *backdoor password*. Tentu saja, mem*bypass* suatu kata sandi boleh berarti mem*boot* kembali sistem, yang mungkin tidak diinginkan. Kemungkinan yang lain akan dicoba untuk menangkap kata sandi melalui jaringan atau kendali *host-based* ( misalnya, *packet sniffer, keystroke logger*), dengan manajemen yang sesuai dan persetujuan yang sah.

### 4.3.3 Menganalisa Data

Suatu analyst's toolkit perlu berisi berbagai tool yang menyediakan kemampuan untuk melaksanakan tinjauan ulang data secara cepat seperti halnya analisa yang dilakukan secara mendalam (detil). Banyak produk mengijinkan analis untuk melaksanakan suatu cakupan luas proses untuk meneliti aplikasi dan file, seperti halnya memperoleh file, membaca gambaran disk, dan mengekstrakdata dari file. Kebanyakan produk analisa juga menawarkan

kemampuan untuk menghasilkan laporan dan untuk membukukan semua kesalahan yang terjadi sepanjang analisa itu.

Walaupun produk seperti itu tidak dinilai di dalam melakukan analisa, pemahaman proses apa yang harus dijalankan untuk menjawab pertanyaan tertentu tentang data adalah suatu langkah pertama yang penting. Suatu analis mungkin perlu menyediakan suatu tanggapan yang cepat atau hanya menjawab suatu pertanyaan sederhana tentang data yang diperoleh tersebut. Di dalam kasus ini, suatu analisa lengkap mungkin diperlukan atau tidak. Sebagai hasilnya, suatu analisa toolkit perlu berisi aplikasi yang dapat memenuhi analisa data dalam berbagai cara dan dapat dijalankan dengan cepat dan efisien dari diskette, CDs, atau analyst workstation. Daftar berikut menyebutkan beberapa jenis proses dimana seorang analis harus bisa melaksanakannya dengan berbagai tool:

1. Penggunaan File *Viewers*. Penggunaan *viewers* sebagai ganti aplikasi sumber yang asli untuk menamppilkan isi dari file jenis tertentu adalah suatu teknik penting untuk membaca sekilas atau mengadakan pra pertunjukan data sebelum dikumpulkan, dan lebih efisien ( misalnya, tidak memerlukan aplikasi asli untuk mengamati masing-masing jenis file). Berbagai *tool* tersedia untuk melihat jenis umum suatu file, dan di sana ada juga tools khusus yang semata-mata untuk mengamati grafik. Jika file *viewers* yang tersedia tidak mendukung format file tertentu , kemudian aplikasi sumber yang asli harus digunakan; jika hal itu tidak tersedia, maka hal tersebut mungkin diperlukan untuk mencari format file dan secara manual mengekstrak data dari file.
2. File yang tidak dikompres.

   File yang dikompres dapat berisi file dengan informasi bermanfaat, seperti file lainnya yang dikompres. Oleh karena itu, hal tersebut sangatlah penting untuk menempatkan analis dan mengekstrak file yang dikompres. File yang tidak dikompres harus dilakukan di awal proses untuk memastikan isi dari file yang dikompres adalah termasuk dalam pencarian dan tindakan lain. Analis perlu mengingat-ingat file yang dikompres yang mungkin berisi *malicious content*, seperti *compression bombs*, yang mana file telah berulang-kali dikompres, *typically dozens* atau *hundreds of times*. *Compression bombs* dapat menyebabkan *tools* untuk pengujian gagal atau mengkonsumsi sumber daya yang pantas dipertimbangkan; mereka dapat juga berisi *malware* dan *malicious payloads*

lainnya. Walaupun tidak ada batasan cara untuk mendeteksi hal utama ini untuk tidak di kompres, dampak mereka mungkin diperkecil. Sebagai contoh, sistem pengujian perlu menggunakan perangkat lunak antivirus terbaru dan harus berdiri sendiri untuk membatasi efek ke sistem itu. Juga, suatu gambaran menyangkut sistem pengujian harus diciptakan sedemikian sehingga jika diperlukan, sistem dapat dikembalikan.

3. Mempertunjukkan Struktur Direktori secara Grafik. Hal ini lebih cepat dan lebih mudah untuk analis mengumpulkan keterangan umum tentang isi suatu media, seperti jenis perangkat lunak yang diinstal dan seperti keserasian teknis dari user adalah membuat data. Kebanyakan produk dapat menampilkan Windows, Linux, dan struktur direktori Unix, selagi produk lain dikhususkan untuk direktori struktur Macintosh.
4. Mengidentifikasi File yang dikenal. Manfaat dari menemukan file yang menarik jelas nyata, tetapi hal itu sering bermanfaat untuk menghapus file tak penting dari pertimbangan, seperti yang dikenal baik yaitu OS dan file aplikasi. Analis dapat menggunakan *hash* untuk menetapkan yang diciptakan oleh rancangan NIST adalah National Software Reference Library (NSRL) atau secara pribadi menciptakan *hash sets* sebagai basis untuk mengidentifikasi *known benign* dan *malicious files. Hash sets* menetapkan secara khusus untuk menggunakan algoritma SHA-1 dan MD5 untuk menetapkan nilai intisari pesan untuk masing-masing file yang dikenal.
5. Melakukan Pencarian Titik Temu dan Mencocokkan Pola. Pencarian Titik temu menopang ketika membaca dengan teliti sejumlah data yang besar untuk temukan kata kunci atau titik temu. Berbagai tool pencarian yang tersedia itu dapat menggunakan *Boolean, fuzzy logic, konsep dan sinonim, stemming,* dan metode pencarian lain. Contoh pencarian umum meliputi pencarian berbagai kata-kata di dalam file tunggal dan pencarian kata-kata tertentu yang salah versi mengejanya. Menetapkan pembangunan secara singkat dari satuan terminologi pencarian untuk situasi umum yang dapat menopang analis di dalam mengurangi volume informasi untuk meninjau ulang. Sebagai tambahan terhadap kepemilikan format file yang tidak bisa dicari titik temunya tanpa *tool* tambahan, dikompres, dienkripsi dan file *password-protected* yang memerlukan pra-proses tambahan sebelum pencarian

titik temu. Penggunaan set *multi-character* data itu meliputi karakter *Unicode* atau asing dapat menyebabkan permasalahan dengan pencarian titik temu; beberapa *tool* pencarian mencoba untuk memperdaya ini dengan menyediakan fungsi penterjemah bahasa. Isu lain yang mungkin adalah batasan *inheren* dari algoritma atau alat pencarian. Sebagai contoh, suatu pencocokkan tidak sampai ditemukan untuk suatu pencarian titik temu jika bagian dari titik temu terletak pada suatu *cluster*, dan sisa dari titik temu berada pada *cluster* yang tidak bersebelahan. Yang dengan cara yang sama, beberapa *tool* pencarian dapat melaporkan suatu pencocokkan yang salah jika bagian dari pencarian suatu titik temu terletak pada satu *cluster* dan sisa dari titik temu terdapat pada *cluster* yang lain dimana bukan bagian dari file yang sama yang berisi *cluster* pertama tersebut.

6. Mengakses File *Metadata*. File *Metadata* menyediakan detil tentang file apapun diberi. Sebagai contoh, memperoleh *metadata* pada suatu file grafis yang mungkin menyediakan grafik tanggal yang dibuat, informasi hak cipta, uraian, dan identitas pembuat. *Metadata* untuk grafik yang dihasilkan oleh suatu kamera digital mungkin meliputi buatan dan model dari kamera digital yang digunakan untuk mengambil gambaran, seperti halnya *F-Stop*, kilat, dan menentukan lubang bidik kamera. Untuk file pengolah kata, *metadata* bisa menetapkan pengarang, organisasi yang diizinkan perangkat lunaknya, kapan dan siapa yang terakhir mengedit, dan komentar *user-defined*. Kegunaan khusus dapat mengekstrak metadata dari file.

Aspek penting lainnya dari meneliti data adalah menguji waktu sistem dan waktu dari file. Mengetahui kapan suatu peristiwa terjadi, suatu file diciptakan atau dimodifikasi, atau suatu e-mail dikirim dapat menjadi kritis untuk analisa data. Sebagai contoh, informasi seperti itu dapat digunakan untuk merekonstruksi suatu batas waktu dari suatu aktifitas. Selagi itu dapat terlihat seperti suatu tugas sederhana, hal itu sering diperumit oleh pertentangan disengaja atau tak disengaja pada pengaturan waktu antar sistemnya. Mengetahui waktu, tanggal, dan wilayah waktu yang menentukan untuk komputer siapa yang datanya akan dianalisa dapat sangat membantu suatu analis; Bagian 5 menguraikan ini secara lebih detil.

Hal itu pada umumnya berpengaruh baik bagi analis jika suatu organisasi memelihara sistemnya dengan *timestamping* akurat. Protokol Waktu Jaringan ( NTP) mensinkronkan

waktu pada suatu komputer dengan suatu jam atomik yang bergerak oleh NIST atau organisasi lain. Sinkronisasi membantu ke arah yang memastikan bahwa masing-masing sistem memelihara suatu pengukuran waktu yang akurat.

Jika berbagai *tool* digunakan untuk analisa yang lengkap, analis perlu memahami bagaimana masing-masing alat untuk mengekstrak, memodifikasi, dan memperlihatkan modifikasi dari file, akses, dan waktu ( MAC) ciptaan. Sebagai contoh, beberapa tool memodifikasi waktu akses terakhir dari suatu direktori atau file jika *filesystem* telah diakhiri dengan menulis ijin oleh sistem operasi itu. *Write-Blockers* mungkin digunakan untuk mencegah *tool* ini untuk memodifikasi waktu MAC. Bagaimanapun, walaupun *write-blockers* dapat mencegah waktu MAC dari yang sedang dimodifikasinya pada media, mereka tidak bisa mencegah sistem operasi dari menyembunyikan perubahan di dalam memori ( yaitu., menyimpan perubahan di dalam RAM). Sebagai konsekuensinya, sistem operasi boleh melaporkan waktu MAC yang disembunyikan daripada waktu MAC yang nyata, dengan demikian mengembalikan hasil yang tidak akurat. Analis harus sadar bahwa waktu akses direktori dan file data terakhir mungkin berubah diantara *query*, tergantung pada alat yang digunakan untuk melaksanakan *query* itu. Oleh karena isu ini, analis perlu berhati-hati untuk memilih MAC yang mengamati metode dan merekam detil dari metode itu .

Dalam banyak kesempatan, analisa tidak hanya melibatkan data dari file, tetapi juga data dari sumber lainnya , seperti sistem operasi status, lintasan jaringan, atau aplikasi. Bagian 8 menyediakan contoh bagaimana data dari file dan data dari sumber lainnya dapat dihubungkan melalui analisa.

## 4.4 Rekomendasi

Rekomendasi utama yang diperkenalkan di dalam bagian ini adalah untuk menggunakan data dari file data diringkas di bawah ini.

1. Analis perlu bekerja dengan salinan file, bukan file yang asli. Sepanjang tahap didapatnya, analis perlu membuat satu atau lebih salinan tentang file yang diinginkan atau sistem file. Analis kemudian bisa bekerja dengan suatu salinan file tanpa mempengaruhi yang asli. Suatu *physical backup* harus dilakukan jika pemeliharaan waktu suatu file adalah hal penting. Suatu *logical backup* sudah cukup untuk mendapatkan suatu file yang informal dari suatu *live systems*.

2. Analis perlu memelihara dan membuktikan integritas suatu file. Penggunaan suatu *write-blocker* selama *backup* mencegah suatu komputer dari suatu penulisan ke media penyimpanannya. Integritas dari data dicopy harus dibuktikan dengan menghitung dan membandingkan pesan inti dari file. *Backups* harus diakses sebagai *read-only* di saat kapan saja yang mungkin; *write-blockers* dapat juga digunakan untuk mencegah penulisan kepada file gambaran atau gambaran yang dikembalikan.
3. Analis perlu mempercayakan *file header* untuk mengidentifikasi tipe isi file. Sebab para user dapat memberikan ekstensi file manapun kepada suatu file, analis mestinya tidak berasumsi bahwa ekstensi file akurat. Analis dapat secara pasti mengidentifikasi jenis data yang disimpan di dalam banyak file dengan memperhatikan *file header* mereka.
4. Analis perlu mempunyai suatu *toolkit* untuk pengujian data. Dimana perlu berisi berbagai tool yang menyediakan kemampuan untuk melaksanakan tinjauan ulang data secara cepat seperti halnya analisa mendetil. *Toolkit* perlu mengijinkan aplikasinya untuk dijalankan dengan cepat dan secara efisien dari media dapat dipindahkan ( misalnya, *diskette*, CD) atau *analysis workstation*.

# Penggunaan Data dari Sistem operasi

# 5

Suatu sistem operasi (OS) adalah suatu program yang dijalankan pada suatu komputer dan menyediakan suatu *platform* perangkat lunak dimana program lain dapat dijalankan. Sebagai tambahan, suatu OS bertanggung jawab untuk mengolah perintah masukan dari seorang pemakai, mengirimkan keluaran ke layar, saling berinteraksi dengan alat penyimpanan untuk menyimpan dan mendapatkan data kembali, dan pengendalian perangkat lain seperti alat pencetak dan *modems*. Beberapa OS umum untuk server atau *workstation* meliputi berbagai versi Windows, Linux, Unix, dan Mac OS. Beberapa alat jaringan, seperti routers, mempunyai OS milik mereka sendiri S ( seperti, *Cisco Internetwork Operating System* [ IOS]). *Personal digital assistants* ( PDA) sering dijalankan oleh sistem operasi khusus, termasuk Palmos dan Windows CE.41 Banyak sistem yang bersifat *embedded* seperti telepon selular, kamera digital, dan *audio players* menggunakan sistem operasi. Bagian ini mendiskusikan komponen dari suatu OS yang mungkin relevan untuk analisa data, dan menyediakan bimbingan untuk melakukan sesuatu dengan sistem operasi server dan *workstation* umum .

## 5.1 Dasar OS

Data OS ada di dalam kedua bagian volatil dan non volatil. Data non volatil mengacu pada data yang tetap berlaku bahkan setelah suatu komputer dimatikan, seperti disimpannya suatu sistem file pada *hard drive*. Data volatil mengacu pada data pada suatu tempat dalam sistem yang hilang setelah suatu komputer dimatikan, seperti koneksi jaringan yang sekarang untuk dan dari sistem itu. Dari suatu analisa yang perspektif, banyak jenis data yang volatil dan non volatil yang mungkin menarik. Bagian ini biasanya mendiskusikan penggunaan dari jenis data OS.

### 5.1.1 Data Non Volatil

Sumber utama data non volatil di dalam suatu OS adalah sistem file. Sistem file pada umumnya adalah sumber data yang paling kaya dan yang paling besar di dalam OS,

kebanyakan berisi dari informasi yang dipulihkan selama pengujian khusus. Sistem file menyediakan penyimpanan untuk OS pada satu atau lebih media. Suatu sistem file secara khusus berisi banyak jenis file yang berbeda, yang mana masing-masing mungkin bernilai untuk analis di dalam situasi yang berbeda. Juga, seperti dicatat di dalam Bagian 4.1.2, data penting yang merupakan sisa juga dapat dipulihkan dari ruang sistem file yang tak terpakai. Daftar berikut menguraikan beberapa jenis data yang biasanya ditemukan di dalam sistem file OS:

1. File Konfigurasi. OS dapat menggunakan file konfigurasi untuk menyimpan OS dan menyiapkan aplikasi. Sebagai contoh, file konfigurasi bisa mendaftarkan layanan untuk dimulai secara otomatis setelah sistem di*boot*, dan menetapkan penempatan *log files* dan file yang bersifat sementara. Para *user* juga dapat memiliki OS sendiri dan aplikasi file konfigurasi yang berisi pilihan dan informasi spesifik untuk *user*, seperti pengaturan yang terkait dengan perangkat keras ( misalnya, resolusi dari layar, setingan printer) dan asosiasi file. File konfigurasi dari bagian tertentu yang menarik adalah sebagai berikut:
    a. Para user dan Kelompok pemakai. OS menyimpan suatu catatan tentang tanggung jawab dari para *user* dan kelompok. Informasi dari *account* ini dapat meliputi keanggotaan kelompok, uraian dan nama *account*, ijin *account*, status *account* ( misalnya, aktif, ditiadakan), dan alur ke direktori utama dari *account*.
    b. File Sandi. OS boleh menyimpan kata sandi yang merupakan bagian dari file data. Berbagai kegunaan *passwordcracking* mungkin digunakan untuk mengkonversi suatu bagian kata sandi menjadi kata sandi sebenarnya yang mengosongkan teks yang sama untuk OS tertentu.
    c. Pekerjaan Yang dijadwalkan. OS memelihara daftar tugas yang dijadwalkan supaya dapat dilakukan secara otomatis pada waktu tertentu ( misalnya, melaksanakan pemeriksaan virus tiap minggu). Informasi yang dapat menarik kesimpulan ini meliputi nama tugas, program yang digunakan untuk melaksanakan tugas, argumentasi dan tombol perintah umum, waktu dan hari manakala tugas dilakukan.
2. *Log*. OS membukukan file berisi informasi tentang berbagai peristiwa sistem operasi, dan dapat juga menjaga informasi peristiwa dari aplikasi yang spesifik. Tergantunglah pada OS, log mungkin disimpan di dalam file teks, *proprietary*-format file biner, atau

database. Juga, beberapa OS menulis masukan *log* untuk dua atau lebih file yang terpisah. Jenis informasi secara khusus yang ditemukan di dalam *log* OS adalah sebagai berikut:

a. Peristiwa Sistem. Peristiwa sistem adalah tindakan operasional yang dilakukan oleh komponen OS, seperti sistem yang dimatikan atau memulai suatu layanan. Secara khas, peristiwa yang digagalkan dan peristiwa paling penting yang sukses dibukukan, tetapi banyak sistem operasi mengijinkan pengurus sistem untuk menetapkan jenis peristiwa mana yang akan dibukukan. Detil catatan untuk masing-masing peristiwa juga bertukar-tukar secara luas; masing-masing peristiwa pada umumnya disertai dengan tanda sesuai waktu terjadinya peristiwa, dan informasi pendukung lain bisa meliputi kode peristiwa, kode status, dan nama user.
b. Arsip Audit. Arsip audit berisi informasi peristiwa keamanan seperti sukses dan tidaknya usaha pengesahan dan perubahan kebijakan keamanan. OS secara khusus mengijinkan pengurus sistem untuk menetapkan jenis peristiwa mana yang harus teraudit. Pengurus juga dapat mengatur beberapa OS untuk pencatatan yang sukses , yang gagal, atau semua usaha untuk melaksanakan tindakan tertentu.
c. Peristiwa Aplikasi. Peristiwa aplikasi adalah tindakan operasional penting yang dilakukan oleh aplikasi, seperti aplikasi *startup* dan *shutdown*, kegagalan aplikasi, dan perubahan konfigurasi dari aplikasi utama. Bagian 7 lebih berisi tentang informasi atas permohonan pencatatan peristiwa.
d.. *Command History*. Beberapa sistem operasi mempunyai log file yang terpisah ( secara khusus untuk masing-masing pemakai) dimana berisi suatu sejarah (menyangkut) perintah OS yang dilakukan oleh masing-masing pemakai.
e. File Yang Baru Diakses. Suatu sistem operasi mungkin mencatat pengaksesan terakhir yang dilakukan pada file atau pemakaian lain, menciptakan daftar file yang diakses yang paling akhir.

3. File Aplikasi. Aplikasi mungkin terdiri atas dari banyak jenis file, termasuk *executables*, skrip, dokumentasi, file konfigurasi, *log file*, history files, grafik, suara, dan ikon.Bagian 7 menyediakan suatu diskusi file aplikasi secara lebih mendalam.

4. File Data. File data menyimpan informasi untuk aplikasi - aplikasi; contoh tentang file data umum meliputi file teks, pengolah kata dokumen, *spreadsheet,* database, file audio, dan file grafik. Juga, kapan data dicetak, kebanyakan sistem operasi menciptakan satu atau lebih file cetakan sementara yang berisi versi *print-ready* dari data. Bagian 4 dan 7 mendiskusikan file data aplikasi secara lebih detil.
5. *Swap Files*. Kebanyakan OS menggunakan *swap file* bersama dengan *random access memory* (RAM) untuk menyediakan penyimpanan sementara untuk data yang sering digunakan oleh aplikasi. Pada dasarnya swap file memberikan jumlah memori yang tersedia untuk suatu program dengan membiarkan halaman ( atau segmen) tentang data untuk ditukar keluar dan masuk RAM. *Swap file* dapat berisi suatu jangkauan luas OS dan informasi aplikasi seperti ID login, kata sandi *hashes*, dan kontak informasi. Bagian 5.1.2 mendiskusikan isi memori secara lebih detil.
6. File Tempat Sampah. Beberapa OS menawarkan kemampuan untuk menyimpan isi memori secara otomatis selama kondisi kesalahan untuk membantu di dalam *troubleshooting* berikutnya. File yang memegang isi memori yang disimpan dikenal sebagai file tempat sampah (*dump file*).
7. *File Hibernasi*. Suatu file hibernasi diciptakan untuk memelihara status yang sekarang dari suatu sistem ( yang secara khusus adalah laptop) dengan perekaman memori dan membuka file sebelum menutup sistem itu. Ketika sistem dinyalakan setelahnya, status dari sistem dikembalikan.
8. File Sementara. Sepanjang instalasi dari suatu sistem operasi, aplikasi, OS atau aplikasi yang diperbaharui dan tingkatkan, file sementara sering dibuat; walaupun file seperti itu secara khususnya dihapus pada ujung proses instalasi, hal ini tidak selalu terjadi. File sementara juga diciptakan ketika banyak aplikasi yang harus dijalankan lagi, file seperti itu harus dihapus ketika aplikasi diakhiri, tetapi hal ini tidak selalu terjadi. File sementara bisa berisi salinan dari file yang lain pada sistem, data aplikasi, atau informasi lain.

Walaupun sistem file adalah sumber yang utama dari data non volatil, sumber data menarik lainnya adalah adalah Sistem Input Output Dasar (*Basic Input/Output System*, atau sering dikenal dengan BIOS). BIOS berisi banyak jenis informasi yang terkait dengan perangkat keras, seperti alat yang dipasang ( misalnya., CD-ROM *drives, hard drives*), jenis

koneksi dan *interrupt request line* ( IRQ) *assignments* ( misalnya, serial, USB, kartu jaringan), komponen - komponen motherboard ( misalnya, tipe dan kecepatan dari prosesor, *cache size*, informasi memori), menentukan sistem keamanan, dan *hot keys* . BIOS juga berkomunikasi dengan RAID drivers dan menampilkan informasi yang disajikan oleh drives itu. Sebagai contoh, BIOS memandang perangkat keras RAID sebagai single drive dan perangkat lunak RAID sebagai *multiple drives*. BIOS secara khas mengijinkan user untuk menetapkan kata sandi, yang mana membatasi akses pengaturan BIOS dan dapat mencegah sistem dari *booting* tanpa kata sandi . BIOS juga menjaga sistem waktu dan tanggal.

### 5.1.2 Data Volatil

Menjalankan OS di dalam RAM dari suatu sistem. Ketika OS sedang berfungsi, isi dari RAM secara konstan berubah. Di setiap waktu, RAM dapat berisi banyak jenis informasi dan data yang mungkin menarik. Sebagai contoh, RAM sering berisi data yang diakses baru – baru ini dan data periodik, seperti file data, *password hashes*, dan perintah terbaru. Juga, serupa ke sistem file, RAM dapat juga berisi data yang bersifat sisa di dalam slack dan *free space*, sebagai berikut:

1. *Slack Space*. Memori *slack space* adalah faktor dominan yang sangat sedikit dibanding file dari *slack space*. Sebagai contoh, suatu OS biasanya mengatur memori di dalam unit yang dikenal sebagai halaman atau blok, dan mengalokasikannya untuk meminta aplikasi di dalam unit tersebut. Kadang-kadang suatu aplikasi tidak dapat meminta keseluruhan dari suatu unit, tetapi hal itu sesekali diberikan. Jadi, unit yang mungkin bersifat data sisa dapat berada dalam unit memori yang dialokasikan untuk suatu aplikasi, walaupun hal itu mungkin tidak bisa dialamatkan oleh aplikasi itu. Untuk efisiensi dan dayaguna, beberapa sistem operasi bertukar-tukar ukuran dari unit yang mereka alokasikan, yang mana di tujukan untuk mengakibatkan ukuran memori slack yang lebih kecil.
2. *Free Space*. Halaman memori yang dialokasikan dan tidak dialokasikan seperti himpunan file. Ketika mereka tidak dialokasikan, halaman memori sering dikumpulkan ke dalam suatu kelompok umum dari halaman – halaman yang tersedia, prosesnya dikenal sebagai *garbage collection*. Hal itu tidak luar biasa untuk data yang bersifat sisa untuk berada di *reusable memory pages*, seperti sekilas tidak mengalokasikan file *clusters*.

Daftar berikut meliputi sebagian dari jenis yang penting lainnya dari data volatil yang mungkin hadir di dalam suatu OS:

1. Konfigurasi Jaringan. Walaupun banyak unsur-unsur *networking,* seperti drives Kartu Penghubung Jaringan ( Network Interface Card atau NIC) dan tetapan konfigurasi, secara khusus disimpan di dalam sistem file, networking adalah hal yang dinamis secara alami. Sebagai contoh, banyak host yang ditugaskan mengalamatkan IP secara dinamis oleh host lainnya , arti dari mengalamatkan IP mereka adalah tidak adanya bagian dari konfigurasi yang disimpan. Banyak host juga mempunyai berbagai alat penghubung jaringan yang ditetapkan, seperti *wired, wireless*, VPN, dan modem; konfigurasi jaringan yang sekarang menandai adanya alat penghubung yang sekarang ini digunakan. Juga, para user mungkin mampu mengubah konfigurasi dari penghubung jaringan dari kelalaian, seperti mengubah alamat IP secara manual.

   Maka, analis perlu menggunakan konfigurasi jaringan yang sekarang, bukan konfigurasi yang disimpan, jika memungkinkan.
2. Hubungan Jaringan. OS memudahkan koneksi antar sistem dan sistem lain. Kebanyakan OS dapat menyediakan daftar koneksi jaringan yang baru keluar dan masuk, dan beberapa OS juga dapat mendaftarkan koneksi yang baru. Karena koneksi yang masuk, OS secara khas menandai sumber daya mana yang sedang digunakan, seperti *printers* dan file bersama. Kebanyakan OS dapat juga menyediakan daftar *port* dan IP yang menunjukkan di mana sistem sedang melakukan pencarian untuk koneksi. Bagian 6 menyediakan suatu pengujian secara mendalam dari arti koneksi jaringan.
3. Menjalankan Proses. Proses adalah program yang sedang dijalankan pada suatu komputer. Proses meliputi layanan yang ditawarkan oleh OS dan aplikasi yang dijalankan oleh *administrator* dan *user*.
4. Kebanyakan OS.menawarkan suatu cara untuk melihat daftar proses yang dijalankan sekarang ini. Daftar ini dapat diperiksa untuk menentukan layanan yang aktif pada sistem, seperti suatu server jaringan, dan program yang digunakan sendiri oleh para *user* ( misalnya, kegunaan *encryption*, pengolah kata, klien e-mail). Daftar proses dapat juga menandai pilihan perintah mana yang digunakan, seperti yang diuraikan di dalam Bagian

7. Mengidentifikasi proses yang sedang berjalan juga sangat menolong untuk mengidentifikasi program yang harus dijalankan tetapi telah tidak diaktifkan atau dipindahkan, seperti perangkat lunak antivirus dan firewalls.

5. File Terbuka. OS boleh memelihara daftar file terbuka, yang mana secara khusus meliputi proses yang membuka masing-masing file atau pemakai.
6. Sesi Login. OS secara khusus memelihara informasi tentang *logged-in* yang sekarang ini dari para *user*( dan jangka waktu dan waktu start dari tiap sesi), gagal *logons* dan sukses sebelumnya, pemakaian yang diistimewakan, dan *impersonation*. Bagaimanapun, informasi sesi login mungkin tersedia hanya jika komputer telah diatur ke usaha logon audit. Arsip *logons* dapat membantu menentukan kebiasaan pemakaian komputer seorang *user* dan mengkonfirmasikan apakah tanggung jawab seorang *user* adalah aktif ketika suatu peristiwa tertentu terjadi.
7. Waktu Sistem Operasi. OS memelihara waktu sekarang dan persiapan waktu yang ditunjukkan dan informasi wilayah waktu. Informasi ini dapat bermanfaat ketika membangun batasan waktu suatu peristiwa atau peristiwa yang menghubungkan antar sistem yang berbeda. Analis harus sadar bahwa waktu yang diperkenalkan oleh sistem operasi mungkin berbeda dengan BIOS dalam kaitannya dengan pengaturan OS-SPECIFIC seperti wilayah waktu.

## 5.2 Memperoleh Data OS

Sebagaimana yang diuraikan di dalam Bagian 5.1, data OS ada di dalam kedua bagian volatil dan non volatil. Data OS seperti data sistem file dapat diperoleh dengan menggunakan pendekatan yang dibahas di dalam Bagian 4 untuk melakukan *logical* and *physical backups*. Data OS yang volatil harus dikumpulkan sebelum komputer dimatikan. Bagian 5.2.1 dan 5.2.2 secara berturut-turut menyediakan rekomendasi untuk memperoleh data OS yang volatil dan non volatil. Bagian 5.2.3 mendiskusikan isu teknis yang dapat menghalangi pengadaan data.

### 5.2.1 Memperoleh Data OS yang Volatil

Data volatile OS menyertakan suatu peristiwa yang dapat diperoleh hanya dari suatu tempat sistem yang belum *rebooted* atau dimatikan sejak peristiwa terjadi. Tiap-Tiap tindakan yang dilakukan pada sistem, apakah inisiatif seseorang atau oleh OS sendiri, akan hampir bisa dipastikan mengubah data OS yang bersifat volatil dalam berbagai cara. Oleh karena itu, analis perlu memutuskan secepat mungkin jika data OS yang bersifat volatil diperlukan untuk

disimpan. Idealnya, ukuran-ukuran untuk pembuatan keputusan ini harusnya telah didokumentasikan sebelumnya sedemikian sehingga analis dapat membuat keputusan yang terbaik dengan seketika. Pentingnya keputusan ini tidak bisa cukup ditekankan, sebab mematikan sistem ataupun memutuskan hubungan sistem dari suatu jaringan dapat menghapuskan kesempatan untuk memperoleh informasi yang berpotensi penting. Sebagai contoh, jika seorang user yang baru saja menjalankan *tool encryption* untuk mengamankan data, RAM pada komputer dapat berisi *password hashes*, yang mana bisa digunakan untuk menentukan kata sandi itu.

Pada sisi lain, mengumpulkan data OS yang bersifat volatil dari suatu komputer yang sedang dijalankan akan tidak bisa dipisahkan dari resiko. Sebagai contoh, kemungkinan selalu ada file itu pada komputer yang dapat berubah dan data OS lain yang bersifat volatil mungkin diubah. Sebagai tambahan, suatu *malicious party* mungkin telah menginstall *rootkits* yang mana dirancang untuk mengembalikan informasi yang palsu, menghapus file, atau melaksanakan tindakan jahat lainnya. Oleh karena itu, resiko dihubungkan dengan kumpulan data OS yang bersifat volatil harus ditimbang potensinya untuk pemulihan informasi penting untuk menentukan jika usaha dijamin. Jika suatu tempat sistem dalam keadaan tidur atau mempunyai perlindungan kata sandi yang terlihat, analis harus memutuskan ya atau tidaknya mengubah status dari sistem dengan mencoba membangunkannya atau berusaha untuk menyusup atau mem*bypass* perlindungan kata sandi, sedemikian sehingga analis dapat mencoba untuk mengumpulkan data yang bersifat volatil.

Usaha yang diperlukan untuk mengumpulkan data volatil tidak sesuai dalam beberapa kasus, maka kekuatan analis sebagai gantinya memutuskan untuk melakukan penonaktifan komputer, sepeti yang diuraikan di dalam Bagian 5.2.2.

Ketika mengumpulkan data OS yang volatil, semua *tool* yang dapat digunakan harus ditempatkan pada suatu disket, CDROM, atau *USB flash drive*, dari *tool* yang mana harus dieksekusi. Setelah itu diijinkan untuk data OS dikumpulkan selagi menyebabkan paling sedikit jumlah gangguan kepada sistem itu. Juga, hanya *tool* yang dipercayai yang harus digunakan sejak seorang user mungkin telah menggantikan perintah sistem dengan program yang tidak baik, seperti seseorang memformat suatu *hard-disk* atau mengembalikan informasi yang salah. Bagaimanapun, jika suatu sistem telah secara penuh dikompromikan, hal ini mungkin untuk *rootkits* dan kegunaan yang bersifat tidak baik lain yang telah diinstall untuk

mengubah kemampuan sistem di tingkatan inti. Hal ini dapat menyebabkan data yang salah dikembalikan ke semua *level user tool*, sekalipun *tool* yang dipercayai untuk digunakan.

Ketika menciptakan suatu koleksi dari *tool* yang dipercayai, file biner yang dihubungkan secara statis harus digunakan. Seperti file *Executable* yang berisi semua fungsi perpustakaan dan semua fungsi yang merupakan acuan, maka memisahkan DLL dan file pendukung lain tidak diperlukan. Hal ini menghapuskan kebutuhan untuk menempatkan versi yang sesuai DLL pada media alat yang dipercayai dan peningkatan keandalan dari *tool*. Analis perlu mengetahui bagaimana masing-masing alat mempengaruhi atau mengubah sistem sebelum memperoleh data volatil itu. Intisari pesan dari tiap alat harus dihitung dan disimpan dengan aman untuk memverifikasi integritas file. Mungkin saja sangat menolong untuk menempatkan suatu catatan pada media alat yang dapat dijalankan untuk mengambil perintah yang sedang dijalankan, tentang waktu masing-masing perintah yang dieksekusi, dan apa yang keluaran dari tiap perintah nya.

Media berisi *tool* perlu melindunginya dari perubahan. Disket harus *writeprotected* untuk memastikan bahwa tidak ada perubahan yang dibuat kepada tool itu. Ketika penggunaan suatu CD-ROM, CD hanya sekali tulis ( yaitu., CD-R) harus digunakan untuk menyimpan *tool* sejak isi dari suatu CD yang bisa ditulis kembali bisa diubah oleh

Kebutuhan membakar CD pada komputer *user*. Setelah suatu *tool* membakar cd sekali tulis, maka pembakaran disk itu harus sampai selesai, yang mana memastikan tidak adanya tambahan data yang dapat ditulis kedalamnya. Banyak kebutuhan dari membakar CD juga mengijinkan sesi yang ada untuk ditutup. Bagaimanapun, menutup sesi yang sederhana mengakibatkan tidak adanya data tambahan yang akan ditulis ke dalam disk dalam sesi tersebut pada kebutuhan membakar CD; hal itu tidak mencegah data tambahan dari yang sedang ditulis ke dalam disk tersebut dalam sesi yang berbeda (sering dikenal sebagai *multisession disc*). Oleh karena itu, sesi tersbut harus diselesaikan, jangan ditutup ketika membuat *CD toolkit*.

Karena media yang berisi *tools* harus diproteksi dari penulisan apapun, hasil yang diproduksi oleh *tools* tidak dapat ditempatkan ke media *tool*. Analis sering mengarahkan alat output ke *floopy disk*, tetapi lazimnya *drive floopy disk* sebagai alat penghitung menurun. Sebagai hasilnya, metode alternatif dari mengoleksi keluaran telah dikembangkan. Khususnya disiapkan CD dan *USB flash drive* yang berisi lingkungan berbasis Windows atau Linux yang dapat digunakan untuk mengumpulkan keluaran tanpa mengubah bagian dari suatu sistem dan

khususnya keluaran yang langsung ke *USB flash drive* lainnya, *external hard drive* atau media penulisan lainnya.

Daftar berikut adalah beberapa jenis dari data OS yang bersifat volatil dan menjelaskan bagaimana *tools* dapat digunakan dalam pengumpulan masing – masing tipe data:

1. Isi dari memori. Ada beberapa kebutuhan untuk dapat meng*copy* isi dari RAM ke file data dan membantu menganalisa data berikutnya. Pada kebanyakan sistem, hal itu tidak dapat menghindari perubahan yang dilakukan pada RAM ketika menjalankan suatu fungsi yang mencoba membuat salinan dari RAM. Oleh karena itu, tujuannya adalah melakukannya dengan langkah - langkah kecil yang mungkin dapat memperkecil gangguan pada RAM
2. Konfigurasi jaringan. Kebanyakan sistem operasi termasuk kebutuhan didalamnya untuk menampilkan konfigurasi jaringan yang ada pada saat itu, seperti *ifconfig* pada sistem Unix dan *ipconfig* pada sistem Windows. Informasi tersebut dapat disediakan melalui kebutuhan dari konfigurasi jaringan dimana dapat meliputi *hostname*, penghubung jaringan secara fisik dan logika dan informasi konfigurasi untuk masing – masing penghubung (misalnya, alamat IP, alamat MAC, dan status yang sekarang).
3. Koneksi jaringan. Sistem operasi secara khusus menyediakan sebuah metode untuk menampilkan daftar dari koneksi jaringan yang ada. Kedua sistem berbasis Windows dan Unix pada umumnya meliputi program *netstat,* dimana daftar koneksi jaringan terdiri dari sumber dan tujuan alamat IP dan *ports,* dan juga daftar *ports* yang terbuka di masing – masing *interface*. Fungsi bagian ketiga yang tersedia akan dapat ditampilkan tugas dari port untuk masing – masing program. Kebanyakan sistem operasi dapat menampilkan daftar sistem file yang tersusun sedikit demi sedikit, dimana menyediakan banyak informasi yang detil daripada daftar koneksi jaringan. Bagian 6.2.7 menyediakan informasi tambahan dalam mengumpulkan informasi dari koneksi jaringan.
4. Menjalankan proses. Semua sistem berbasis Unix menawarkan perintah *ps* untuk menampilkan proses yang sedang dikerjakan. Walaupun Windows menawarkan kebutuhan daftar proses berbasis GUI , *task manager*, semua hal itu pada umumnya akan lebih baik jika memiliki daftar *text* dasar. Utilitas bagian ketiga dapat digunakan untukmenghasilkan daftar teks dari proses yang dijalankan untuk sistem Windows.

5. File terbuka. Semua sistem berbasis Unix menawarkan perintah *lsof* untuk menampilkan daftar dari file yang terbuka. Manfaat bagian ketiga dapat digunakan untuk menghasilkan daftar teks dari file yang terbuka untuk sistem Windows.
6. Sesi login. Banyak sistem operasi yang memiliki perintah sendiri untuk mendaftar pencatatan *user* komputer saat itu, seperti daftar dari alamat sumber dari masing – masing *user* komputer dan ketika *user* itu tercatat masuk ke dalam suatu sistem. Kegunaan bagian ketiga yang tersedia dapat mencatatkan *user* yang terkoneksi saat itu pada sistem Windows.
7. Waktu sistem operasi. Ada beberapa kegunaan yang dapat digunakan untuk mendapatkan waktu dari sistem yang sekarang digunakan, informasi wilayah waktu dan pengaturan waktu perubahan. Pada sistem Unix, perintah *date* dapat digunakan untuk mendapatkan informasi ini pada sistem Windows, d*ate, time* dan perintah *nlsinfo* dapat digunakan untuk secara bersamaan dalam mendapatkan informasi ini.

Sebagai tambahan untuk *tool* ini, hal ini sering berguna untuk memasukkan *tool* dengan tujuan umum dalam suatu *toolkit,* seperti yang dibawah ini :

1. *Command prompt* suatu OS. Hal ini merupakan suatu fungsi yang menyediakan *command prompt* pada OS melalui *tool* yang lain dalam *toolkit* yang dapat dieksekusi, seperti *cmd* pada sistem Windows.
2. SHA-1 *Checksum*. Fungsi yang dapat menghitung isi pesan SHA-1 dari file data yang sangat membantu dalam memverifikasi file. Hal itu mungkin juga berguna untuk masuk ke dalam *toolkit* yang merupakan daftar dari isi pesan SHA-1 untuk sistem file data yang dihubungkan dengan target OS dalam membantu untuk memverifikasi file. Fungsi tersebut tersedia untuk tujuan ini pada berbagai macam OS .
3. Daftar direktori. Suatu fungsi untuk mendaftar isi dari direktori harus dimasukkan untuk mengendalikan sistem file dam melihat isi dari sistem file. Pada kenyataannya pada semua sistem operasi terdapat fungsi, untuk contohnya, perintah *ls* yang digunakan pada sistem Unix, sedangkan pada sistem Windows mengunakan perintah *dir*.
4. Pencarian Mata Rantai. Sebuah fungsi untuk melakukan pencarian teks dalam string mungkin berguna dalam mengidentifikasi file data yang menarik. Sistem Unix menawarkan perintah *grep* untuk melakukan pencarian teks dalam bentuk string dan fungsi *grep* bagian ketiga juga tersedia dalam sistem Windows

5. Editor Teks. Suatu editor teks sederhana mungkin bermanfaat untuk mengamati file teks atau menggubah catatan. Banyak para editor teks yang tersedia, seperti *Notepad* pada sistem Windows dan *vi* pada sistem Unix.

Jenis dari data volatil yang harus diperoleh dengan *toolkit* bergantung pada kebutuhan yang spesifik. Sebagai contoh, jika ada gangguan jaringan yang dicurigai, lalu kemungkinannya hal itu berguna untuk mengumpulkan informasi konfigurasi jaringan, koneksi jaringan, sesi login dan proses yang dijalankan untuk menentukan bagaimana seseorang memperoleh akses ke suatu sistem. Jika suatu penyelidikan menyangkut spyware, kemudian isi dari RAM, daftar proses yang dijalankan daftar file yang terbuka, informasi konfigurasi jaringan dan koneksi jaringan dapat mengungkapkan keamanan sosial dan nomor kartu, program yang digunakan untuk memperoleh atau mengenkripsi data, potongan kata sandi dan metode yang mungkin telah digunakan untuk mengirim informasi ke luar jaringan. Ketika dalam keraguan, hal itu secara normal merupakan ide yang baik dalam mengumpulkan banyak data volatil ketika memungkinkan sejak semua peluang untuk mendapatkan data akan hilang sama sekali setelah komputer dimatikan. Sebuah penentuan dapat dibuat setelah itu seperti yang mana dikumpulkannya data volatile yang harus diuji. Suatu skrip yang diotomatiskan pada *CD toolkit* dapat digunakan untuk konsistensi dalam mengumpulkan data volatil. Skrip dapat meliputi cara untuk memindahkan informasi yang dikumpulkan ke media penyimpanan lokal, seperti *thumb drive* dan ke lokasi *networked drive.*

Sejak data volatil memiliki kencenderungan untuk bertukar waktu, ketepatan waktu dan pesanan data volatil mana harus diperoleh adalah suatu hal penting. Dalam banyak kasus, pertama analis harus mengumpulkan informasi pada koneksi jaringan dan sesi login, sejak koneksi jaringan sudah diluar waktu batasnya atau tidak tersambung dan daftar user yang terkoneksi ke sistem pada suatu waktu dapat bertukar. Data volatil mungkin lebih sedikit untuk berubah, seperti informasi konfigurasi jaringan, yang harus diperoleh kemudiannya. Pesanan yang direkomendasikan dalam data volatile yang mana secara umum harus dikumpulkan, didaftar dari awal sampai akhir, adalah sebagai berikut :

1. Koneksi Jaringan (*Network Connections*)
2. Sesi Login (*Login Sessions*)
3. Isi Memori (*Contents Of Memory*)
4. Menjalankan Proses (*Running Processes*)

5. File Yang Terbuka (*Open Files*)
6. Konfigurasi Jaringan (*Network Configuration*)
7. Waktu Sistem Operasi(*Operating System Time*.)

### 5.2.2 Mendapatkan data non volatil

Setelah mendapatkan data OS yang volatil, analis sering perlu juga untuk mendapatkan data OS yang non volatil. Untuk melakukannya, pertama analis perlu untuk memutuskan apakah sistem harus dimatikan atau dimatikan atau tidak.hal ini mempengaruhi kemampuan untuk melakukan *physical backups* dan banyak *logical backups,* tetapi dapat juga mengubah data OS yang dipelihara. Kebanyakan sistem dapat dimatikan melalui dua metode, seperti berikut :

1. Melakukan pematian sistem operasi secara baik. Hampir tiap – tiap OS menawarkan pilihan cara mematikan OS . Penyebabnya adalah suatu OS yang melakukan aktifitas *cleanup*, seperti menutup semua file yang terbuka, menghapus file sementara dan kemungkinan membersihkan file *swap*, sebelum mematikan sistem. Mematikan sistem secara baik dapat juga memicu pemindahan materi *malicious*; untuk contohnya, memori *resident rootkits* dapat hilang dan *trojan horses* mungkin memindahkan bukti dari aktifitas buruk mereka. OS secara khususnya dimatikan dari *account administrator* ataupun *user* yang sedang menggunakan sistem (jika user yang sedang menggunakan sistem memiliki cukup perlakuan khusus).
2. Memindahkan tenaga dari sistem. Pemutusan hubungan tenaga dari belakang komputer (dan memindahkan baterai pada laptop atau alat yang dapat berpindah) dapat memelihara file *swap*, file data sementara, dan informasi lainnya yang mungkin diubah atau dihapus selama sistem dimatikan secara baik. Sayangnya, kehilangan tenaga secara tiba – tiba dapat menyebabkan banyak OS untuk merusak data, seperti file yang terbuka. Juga, untuk alat yang banyak digunakan konsumen, seperti PDA dan telepon selular, memindahkan tenaga baterai dapat menimbulkan hilangnya data..

Analis harus sadar akan karakteristik dari tiap sistem operasi dan memilih suatu metode *shutdown* berdasar pada perilaku khusus dari OS dan jenis data yang perlu untuk dipelihara. Sebagai contoh, sistem Windows 95/98 dan dos pada umumnya tidak merusakkan data ketika tenaga listriknya dipindahkan tiba – tiba, jadi walau memindahkan tenaga listrik data harus dapat dilindungi. Sistem operasi lainnya mungkin dapat merusak data, seperti file

yang terbuka atau file yang sedang diakses ketika itu, dari hilangnya tenaga listrik, maka mematikan komputer secara baik merupakan hal yang pada umumnya baik kecuali kalau file *swap* dan file data sementara merupakan bagian yang menarik atau jka sistem berisi *rootkits, trojan horses* atau program tidak baik lainnya yang mungkin disebabkan oleh mematikan sistem secara benar. Setelah melakukan "*shutdown*", analis kemudian perlu memperoleh data sistem file dari media penyimpanan sistem menggunakan metode yang dibahas dalam bagian 4.

Ketika sekali saja data sistem file telah didapatkan, *tool* dapat digunakan untuk mendapatkan jenis data yang spesifik dari sistem file. Mendapatkan file reguler, seperti data, aplikasi dan file konfigurasi merupakan hal yang secara langsung relatif dan diuraikan secara lebih detil di bagian 4. Daftar berikut merupakan jenis lainnya dari data sistem operasi yang bersifat non volatil dan penjelasan bagaimana *tool* dapat berguna dalam mendapatkan masing – masing jenis dari sistem file :

1. Para *user* dan Grup *user*. Sistem operasi memelihara daftar para *user* dan grup *user* yang memiliki akses ke suatu sistem. Pada sistem Unix, para user dan grup user didaftarkan berturut - turut dalam */etc/passwd* dan */etc/groups*. Sebagai tambahannya, perintah para grup dan *user* dapat digunakan untuk mengidentifikasi *user* yang telah dimasukkan kedalam sistem dan grup dimana mereka menjadi anggota. Pada sistem Windows, perintah *user* dan grup *user* jaringan dapat digunakan untuk menyebutkan satu persatu para *user* dan grup *user* pada suatu sistem.
2. Kata sandi. Kebanyakan sistem operasi menjaga potongan (bagian) kata sandi untuk kata sandi *user* pada suatu disk. Pada sistem Windows, fungsi bagian ketiga dapat digunakan untuk membuang potongan kata sandi dari database *Security Account Manager* (SAM). Pada sistem Unix, potongan kata sandi biasanya dalam file the */etc/passwd* atau */etc/shadow*. Seperti yang diuraikan pada bagian 4.3.2, program yang dapat membuka (*cracking*) kata sandi dapat digunakan dalam mencoba mengekstrak kata sandi dari potongannya.
3. Jaringan bersama. Suatu sistem memungkinkan sumber daya lokal untuk dibagi ke jaringan lainnya. Pada sistem Windows, fungsi *SrvCheck* dapat digunakan untuk menyebutkan satu persatu jaringan bersama. Fungsi bagian ketiga dapat menyediakan informasi yang sama untuk sistem operasi lainnya.

4. *Logs. Logs* yang tidak disimpan dalam file teks mengharuskan penggunaan fungsi dari *log* yang diekstrak. Untuk contohnya, fungsi yang dikhususkan dapat memperoleh kembali informasi tentang sukses terbaru dan mencoba menggagalkan "*logon"* pada sistem Windows. Kebanyakan masukan *log* pada sistem Unix disimpan dalam file teks oleh *syslog* atau dalam direktori */var/log/*, jadi tidak perlu fungsi khusus untuk memperoleh informasi dari *log*. Pencarian nama file berakhir. *Log* harus mengidentifikasi kebanyakan file *log.*

Adakalanya, analis perlu untuk mendapatkan data dari BIOS, seperti jenis sistem pengolah waktu dan tanggal, dan kecepatan. Terutama sejak BIOS berisi informasi yang berhubungan dengan konfigurasi perangkat keras, data yang didapatkan dari BIOS kebanyakan seperti hal yang dibutuhkan ketika admin suatu sistem menyelesaikan masalah isu operasional. Secara khususnya, analis membutuhkan data BIOS yang pertama didapat dari data volatil dan sistem file yang dibutuhkan, kemudian "*reboot"* sistem dan serang kunci fungsi yang sesuai (pada umumnya ditetapkan dalam layar inisial selama "*boot*") untuk menampilkan pengaturan dari BIOS. Jika kata sandi BIOS diatur, analis tidak mungkin mendapatkan akses ke pengaturan BIOS secara mudah dan kemungkinan dicoba untuk mengira kata sandi "*default*"nya atau mengelakkan proteksi dari kata sandi. Disini ada berbagai metode yang digunakan untuk melewatkan kata sandi BIOS, termasuk menemukan kata sandi sesuai yang dibuat secara tersembunyi, menggunakan kata sandi " *cracker"* (orang yang sering membuat celah kata sandi) , memindahkan *jumper* yang sesuai pada *motherboard* atau memindahkan baterai CMOS (jika memungkinkan). Masing – masing sistem mungkin dapat berbeda, jadi analis harus mulai meriset karakteristik sistem yang diuraikan dalam dokumentasi motherboard untuk menghindari kerugian suatu sistem yang tidak perlu.

### 5.2.3 Isu Teknis dengan Memperoleh Data

Ada isu teknis yang berpotensi dapat menghalangi perolehan data OS. Bagian 44 menguraikan beberapa isu berhubungan sistem file; bagian ini fokus pada isu didapatkannya tambahan dan penyediaan bimbingan pada hal apa, meskipun terdapat perbedaan, bisa dilakukan untuk mengurangi isu. Tujuan bagian ini adalah tidak menyediakan suatu ikhtisar secara menyeluruh tentang isu yang mungkin, tapi lebih untuk menyediakan informasi pada suatu yang umum.

1. Akses OS. Mendapatkan data volatil mungkin sulit karena analis tidak dapat secara siap untuk memperoleh akses ke sistem operasi. Sebagai contoh, seorang user mungkin menjalankan kata sandi yang melindungi *screensaver* atau memiliki sistem yang terkunci; analis perlu untuk mengelakkan proteksi tersebut dan mencari jalan lain untuk mendapatkan akses ke data OS yang volatil tersebut. Jika kata sandi yang melindungi *screensaver* aktif, mengulang pengaktifan sistem dapat mengijinkan analis untuk melewati *screensaver*, tapi akan juga menyebabkan data OS yang bersifat volatil hilang. Secara kemungkinan yang lain adalah suatu *host* mungkin menggunakan otentikasi berdasar biometrik, seperti pembaca sidik jari, atau layanan tambahan otentikasi lainnya; hal ini bisa menyebabkan isu yang serupa dalam pengaksesan data OS yang bersifat volatil. Ada fungsi bagian ketiga untuk beberapa OS yang diklaim untuk memecakan kata sandi *screensaver* tanpa me"*reboot*" sistem. Fungsi tersebut biasanya bersandar pada fitur otomatis yang dijalankan *drive* suatu CD; fungsi yang secara otomatis berjalan di dalam *background*, kemudian meletakkan kata sandi yang dienkripsi dan mencoba untuk mendekripsikannya.
2. Modifikasi *log*. *User* mungkin mencoba untuk mengurangi penggunaan yang tidak berguna dari *log* dengan cara menon-aktifkan fitur *log*, pengaturan memodifikasi *log* sedemikian sehingga tersedia tempat penyimpanan kecil untuk *log* atau menulis banyak peristiwa palsu ke *log*. Salah satu jalan untuk mengurangi dampak perubahan pencatatan adalah mengkonfigurasi sistem ke arsip *log* masukkannya pada server yang dipusatkan.
3. *Hard Drives* dengan *Flash Memory*. Adakalanya, seorang analis mendapatkan *hard drive* yang juga berisi *flash memory*. *Flash memory* ini dapat berisi kata sandi yang dibutuhkan untuk mengakses *drive*, bahkan ketika *drive* telah dipindahkan dari komputer tersebut. Secara khusus, analis perlu untuk menemukannya, mengira, atau memecahkan kata sandi untuk mendapatkan akses ke *drive* tersebut.
4. *Key remapping*. Pada banyak komputer, kunci perorangan atau kombinasi dari tombol dapat dipetakan kembali untuk melakukan fungsi yang berbeda dari tujuan awal mereka. Untuk contoh, seseorang dapat memetakan kunci *Ctr*l, *Alt* dan *Del* maka mereka menyeka *hard drive* komputer sebagai ganti tindakan yang diharapkan, dalam kata lain me"*reboot*" sistem. Seorang analis yang menggunakan *keyboard* komputer dapat menekan tombol yang menyebabkan tindakan yang tidak diharapkan dilakukan. Maka, jalan terbaik dalam menghindarkan masalah pemetaan kembali suatu kunci adalah dengan mendapatkan data

dari komputer tanpa menggunakan *keyboard*nya. Sebagai contohnya, seseorang analis harus bisa menyertakan *analysis workstation* ke komputer yang diinginkan menggunakan kabel jaringan seberang dan menjalankan skrip dari *workstation.*

## 5.3 Pengujian OS

Data Berbagai teknik dan *tool* dapat digunakan untuk mendukung proses pengujian. Banyak dari hal yang sama dibahas di dalam Bagian 4.3 untuk pengujian data diperoleh file juga dapat digunakan untuk diperolehnya data OS. Juga seperti yang diuraikan dalam bagian 7, aplikasi keamanan seperti pengawas integritas file dan sistem deteksi gangguan berdasarkan host dapat sangat membantu dalam mengidentifikasi aktifitas tidak baik terhadap sistem komputer. Sebagai contohnya, pengawas integritas file dapat digunakan untuk menghitung isi pesan dari file OS dan membandingkannya terhadap basis data yang isi pesannya dikenal untuk penentuan jika ada file yang disepakati. Jika perangkat lunak untuk mendeteksi gangguan telah diinstal di komputer, maka perangkat lunak itu mungkin berisi *logs* yang menandai tindakan yang dilakukan terhadap OS.

Isu lainnya yang berhadapan dengan analis adalah pengujian file *swap* dan sampah RAM , dimana file data biner yang besar mengisi data tidak terstruktur. *Hex Editors* dapat digunakan untuk membuka file ini dan menguji isinya; bagaimanapun, mencoba secara manual untuk menempatkan data yang dapat dimengerti menggunakan *hex editor* pada file yang besar dapat menghabiskan banyak waktu untuk proses. Penyaringan tools mengotomatiskan proses dari pengujian file swap dan sampah RAM dengan mengidentifikasikan pola teks dan nilai yang secara *numeric* dapat berpotensi menghadirkan nomor telepon, nama orang, alamat email, alamat web, dan jenis informasi kritis lainnya.

Analis sering ingin untuk mengumpulkan informasi tambahan pada program tertentu yang dijalankan pada suatu sistem. Setelah mendapatkan daftar proses yang sedang dijalankan pada suatu sistem, analis dapat melihat nama proses untuk mendapatkan informasi tambahan, seperti tujuan dari proses dan pembuatnya. Bagaimanapun, user mungkin mengubah nama dari program untuk merahasiakan fungsinya, seperti menamakan program *trojan* dengan *calculator.exe*. oleh karena itu, proses memeriksa nama harus hanya dilakukan setelah memverifikasi identitas dari memproses file dengan penghitungan dan membandingkan isi pesannya. Pemeriksaan yang sama dapat dilakukan pada file *library*, seperti *dynamic link*

*libraries* (DLL) pada sistem Windows, untuk menentukan *library* mana yang akan diisi dan apa tujuan khususnya.

Seperti yang diuraikan pada bagian 5.2, analis mungkin mendapatkan banyak tipe data OS yang berbeda, termasuk berbagai sistem file. Mencoba menyelidiki sampai masing – masing jenis data untuk mendapatkan informasi yang relevan dapat menjadi proses dengan waktu yang intensif. Analis biasanya menemukannya dan menggunakannya untuk mengidentifikasi beberapa sumber data untuk meninjau ulang dari awal, dan kemudian menemukan sumber lainnya yang seperti itu dari dasar informasi penting pada tinjauan ulang tersebut . Juga dalam banyak kesempatan, analisa dapat melibatkan data dari jenis sumber lainnya, seperti *network traffic* atau aplikasi - aplikasi . Bagian 8 menyediakan contoh bagaimana data dari sistem operasi dan sumber lainnya dapat dikorelasikan melalui analisa.

## 5.4 Rekomendasi

Kunci rekomendasi yang diperkenalkan dalam bagian ini untuk menggunakan data dari sistem operasi diringkas di bawah ini

1. Analis harus bertindak sewajarnya untuk menjaga data OS yang volatil. Kriteria untuk menentukan apakah data OS yang volatil perlu untuk dipelihara harus didokumentasikan terlebih dahulu sehingga analis dapat membuat keputusan yang diinformasikan secepat mungkin. Resikonya dihubungkan dengan pengumpulan data OS yang bersifat volatil harus dipikirkan terhadap potensinya untuk memperoleh kembali informasi untuk menentukan jika usahanya telah dijamin.
2. Analis perlu menggunakan *toolkit* yang dapat dipercaya untuk mendapatkan data OS yang volatil. Membuatnya dapat memberikan data OS yang akurat untuk dikoleksi ketika menyebabkan sedikit jumlah gangguan pada sistem dan menjaga *tools* dari perubahan. Analis haru mengetahui bagaimana masing – masing *tool* dapat mempengaruhi atau mengubah sistem ketika mendapatkan data.
3. Analis perlu memilih metode "*shutdown*" yang sesuai untuk masing – masing sistem. Masing – masing cara dari "*shutting down*" sistem operasi tertentu dapat menyebabkan jenis data yang berbeda untuk dipelihara atau dirusak, maka analis harus sadar tentang jenis – jenis "*shut down"* dari masing – masing OS.

# Penggunaan Data dari Network Traffic

# 6

Analis dapat mengunakan data dari network traffic untuk merekonstruksi dan meneliti serangan dasar jaringan dan penggunaan jaringan yang sesuai, seperti bermacam jenis *troubleshooting* dari masalah operasional. Arti dari *network traffic* mengacu pada komunikasi jaringan komputer yang diganti menjadi tanpa kabel atau diluar kabel diantara *host*. Bagian ini menyediakan perkenalan tentang *network traffic*, termasuk engertian dari sumber utama dari *network traffic data* (seperti, perangkat lunak pendeteksi gangguan, *firewalls*). Hal tersebut nantinya akan mendiskusikan teknik untuk mendapatkan data dari sumber ini dan mendapatkan nilai legal yang berpotensi dan keterangan secara tenis dalam mendapatkan data. Bagian lainnya terfokus ada teknik- teknik dan *tools* yang digunakan untuk menguji data dari *network traffic*. Karena pengetahuan dasar dari *Transmission Control Protocol/Internet Protocol* (TCP/IP) dibutuhkan untuk dapat mengerti tentang data, *tools*, dan metodologi yang disajikandalam bagian ini, hal itu akan dimulai dengan gambaran tentang TCP/IP.

## 6.1 Dasar TCP/IP

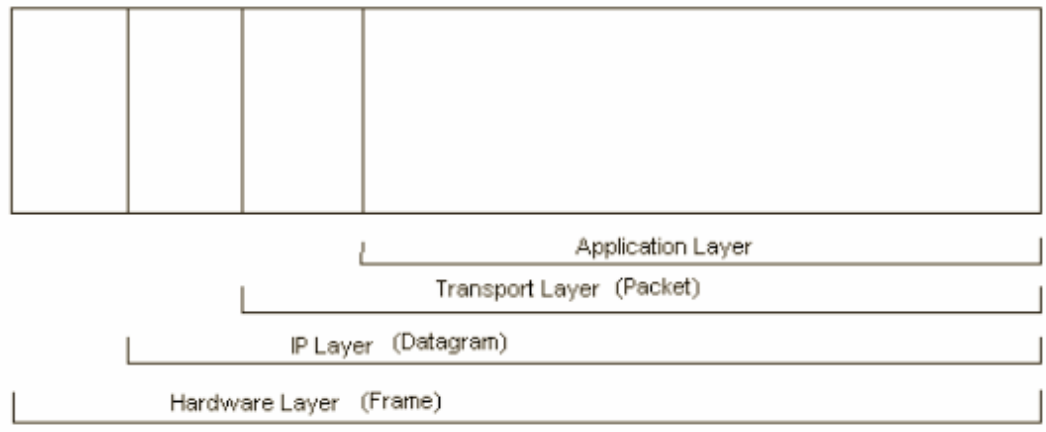
| |
|---|
| *Application Layer*. Lapisan ini mengirimkan dan menerima data untuk aplikasi tertentu, seperti Domain Name System ( DNS), HyperText TransferProtocol ( HTTP), dan Simple Mail Transfer Protocol ( SMTP). |
| *Transport Layer*. Lapisan ini menyediakan jasa tanpa koneksi atau berorientasi koneksi untuk mengangkut jasa lapisan aplikasi antar jaringan. *Transport Layer* dapat secara bebas memilih meyakinkan keandalan komunikasi. Transmission Control Protocol ( TCP) dan User Datagram Protocol ( UDP) biasanya digunakan protokol *transport layer*. |
| *Internet Protocol Layer* (juga dikenal sebagai Lapisan Jaringan). Lapisan ini mengarahkan paket ke seberang jaringan. Internet Protokol (IP) adalah protokol lapisan jaringan pokok untuk TCP/IP. Protokol lain yang biasanya digunakan di lapisan jaringan adalah Internet Control Message Protocol ( ICMP) dan Internet Group Management Protocol ( IGMP). |
| Hardware Layer (juga dikenal sebagai *Data Link Layer*). Lapisan ini menangani komunikasi pada komponen jaringan fisik. *Data Link Layer* terbaik yang dikenal adalah Ethernet. |

Gambar 6.1 Empat Susunan TCP/IP

TCP/IP secara luas digunakan untuk diseluruh dunia untuk menyediakan jaringa komunikasi. Komunikasi TCP/IP disusun dari empat susunan yang dikerjakan bersama -

sama. Ketika pengguna menginginkan untuk memindahkan data ke jaringan lainnya, data dinilai dari susunan tertinggi sampai susunan menengah hingga susunan terendah, dengan masing – masing susunan ditambahkan informasi tambahan. Susunan terendah mengirim akumulasi data melalui *physical network*; kemudian data dinilai naik sampai susunan untuk tujuan akhirnya. Pada dasarnya, data yang diproduksi oleh susunan di enkapsulasi dalam tempat yang besar oleh layer sebelumnya. Empat susunan TCP/IP, dari yang tertinggi sampai yang terendah, diperlihatkan pada gambar 6-1.

Empat susunan bekerja bersamaan untuk memindahkan data antara *host*. Dapat dilihat di gambar 6-2, masing – masing susunan enkapsulasi adalah susunan sebelumnya. Hal berikutnya menggambarkan masing – masing dari susunan dan hasil dari karakteristik yang sangat berhubungan dengan analisis data jaringan. Bagian 6.1.5 menjelaskan bagaimana susunan - susunan saling berhubungan satu dengan lainnya.




Gambar 6.2. Enkapsulasi TCP/IP

### 6.1.1 Susunan aplikasi

Susunan aplikasi memungkinkan aplikasi – aplikasi untuk memindahkan data antara aplikasi *server* dan *client*. Contoh dari protokol susunan aplikasi adalah *HyperText Transfer Protocol* (HTTP), yang mana memindahkan data antara s*erver* web dan *browser* web. Aplikasi protokol susunan biasa lainnya meliputi *Domain Name System* (DNS), *File Transfer Protocol* (FTP), *Simple Mail Transfer Protocol* (SMTP), dan *Simple Network Management Protocol* (SNMP). Disini ada ratusan aplikasi protokol susunan unik yang biasa digunakan dan banyak lagi yang jarang digunakan. Bagaimanapun protokol digunakan, data aplikasi

dihasilkan dan kemudian dinilai untuk susunan pengangkut untuk proses selanjutnya bagian 7 fokus pada memperoleh data aplikasi yang dihubungkan dan pengujian.

### 6.1.2 Susunan pengangkut

Susunan pengangkut bertanggung jawab untuk mengemasi data hingga dapat ditransmisikan antar *host*, setelah susunan pengangkut dienskapsulasi data aplikasi, *logical units* menghasilkan sesuatu yang mengacu pada paket ( suatu paket dapat juga dibuat tanpa data aplikasi – untuk contohnya, ketika mengadakan hubungan pertam kali) masing – masing paket berisi informasi, yang mana merupakan macam – macam bagian yang digabungkan yang juga merupakan karakter spesifik dari protokol pengangkut yang digunakan dan secara bebas berisi muatan yang menguntungkan, dimana juga menjaga data aplikasi

Kebanyakan aplikasi yang berkomunikasi diluar jaringan bersandar pada susunan pengangkut untuk berusaha memastikan pengiriman data dapat dipercaya. Secara umum, hal ini akan selesai dengan menggunakan *Transmission Control Protocol* (TCP) protokol susunan pengangkut, dimana membuat koneksi diantara dua *host* dan kemudian membuat usaha terbaik untuk memastikan transfer data diluar koneksi dapat dipercaya. Masing – masing paket TCP meliputi *port* sumber dan *port* tujuan. Satu dari *port – port* diasosiasikan dengan aplikasi *server* dalam satu sistem dan *port* lainnya diasosiasikan dengan aplikasi yang mengkoresponden klien pada sistem yang lain. Sistem klien secara khususnya memilih nomor *port* untuk penggunaan aplikasi yang tersedia, ketika sistem *server* secara normal memiliki nomor *port* yang diperuntukkan untuk masing – masing aplikasi. Walaupun banyak *port* server yang secara normal digunakan oleh aplikasi tertentu (misalnya *FTP servers* at port 21, *HTTP servers* at port 80), banyak apliksi server yang dapat dijalankan dari nomor port manapun, jadi hal itu tidak bijaksana untuk mengambil *network traffic* yang berisi data dari dasar aplikasi yang pasti semata – mata pada nomor *port server*.

Ketika kehilangan beberapa data aplikasi yang tidak penting (misalnya, *streaming audio, video*), *User Datagram Protocol* (UDP) yang secara khusus digunakan. Melibatkan UDP mengurangi pengeluaran tambahan dan keterlambatan daripada TCP karena UDP memiliki koneksi yang sedikit; satu *host* mengirim data ke *host* lainnya secara sederhana tanpa mengadakan persiapan. UDP juga digunakan untuk aplikasi yang mau mengambil tanggung jawab untuk memastikan pengiriman data dapat dipercaya, seperti DNS dan aplikasi yang diharapkan hanya untuk penggunaan pada area jaringan local, seperti *Host.*

*Configuration Protocol* (DHCP) dan SNMP. Seperti TCP, masing – masing paket UDP berisi *port* sumber dan *port* tujuan. Walaupun *port* UDP dan TCP sangat mirip, mereka berbeda satu dengan lainnya dan dan tidak dapat tertukar. Banyak aplikasi (seperti DNS) dapat menggunakan kedua *port* TCP dan UDP; meskipun secara khusus aplikasi menggunakan nomor yang sama untuk *port* TCP dan *port* UDP, hal ini dibutuhkan.

### 6.1.3 Susunan IP

Susunan IP dapat juga disebut susunan jaringan, karena hal itu bertanggung jawab untuk menangani pengalamatan dan mengarahkan data yang diterimanya dari susunan pengangkut. Informasi IP berisi kolom yang disebut versi IP, yang mana menunjukkan versi yang mana dari protokol IP yang digunakan. Secara khususnya diatur sampai empat untuk IPv4, tetapi penggunaan IPv6 makin bertambah, maka kolom ini mungkin dapat diatur sampai 6. Disamping kolom versi IP, informasi penting lainnya seperti berikut :

1. Alamat sumber dan tujuan IP. Hal ini adalah tentang alamat “dari” dan “untuk” yang dimaksudkan untuk menunjukkan hasil akhir dari komunikasi. Contohnya alamat IP 10.3.1.70 (IP versi 4) dan 1000:0:0:2F:8A:400:0427:9BD1(IP versi 6).
2. Nomor protokol IP. Hal ini menunjukkan protokol susunan pengangkut yang mana berisi muatan yang mendukung IP. Biasanya digunakan nomor protokol IP yang meliputi 1 (ICMP), 6 (TCP), 17 (UDP), dan 50(*Encapsulating Security Payload* [ESP]).

Susunan IP juga bertanggung jawab untuk menyediakan menyediakan informasi status dan kesalahan melibatkan pengalamatan dan pegarahan data;hal itu dilakukan dengan Internet Control Message Protocol (ICMP). ICMP adalah protokol dengan sedikit koneksi yang tidak membuat jaminan pesan status dan kerusakan untuk dikirimkan. Karena hal itu didesain untuk informasi pemindahan yang terbatas, bukan data aplikasi, ICMP tidak memilki port; sebagai gantinya, dia memiliki jenis pesan, yang mana menunjukkan tujuan dari masing – masing pesan ICMP. Beberapa jenis pesan Destination Unreachable memiliki beberapa kode pesan yang mungkin yang menunjukkan apa yang dimaksud unreachable (misalnya network, host, protokol). Kebanyakan pesan ICMP tidak diharapkan untuk menimbulkan tanggapan.

Alamat IP seringkali digunakan melalui susunan yang tidak langsung. Ketika orang membutuhkan untuk mengakses sumber pada jaringan, seperti web server atau *e-mail server*, mereka secara khusus mencantumkan nama server, seperti www.nist.gov, daripada alamat server IP. Namanya juga dikenal sebagai nama *domain*, yang dipetakan ke alamat IP melalui

protokol susunan aplikasi *Domain Name System* (DNS). Alasan utama untuk mencantumkan nama domain daripada alamat IP adalah hal itu secara umum lebih memudahkan orang utuk mengingat nama dibanding alamat IP. Juga alamat IP *host* kemungkinan mengubah waktu lagi; dengan mereferensikan *host* oleh nama *domain*, yang mana setelah itu dipetakan ke alamat IP *host, user* dapat menjangkau pengguna *host* dengan nama yang sama tidak masalah dimana *host* sedang menggunakan alamat IP .

### 6.1.4 Susunan perangkat keras

Seperti yang disiratkan namanya, susunan perangkat keras melibatkan komponen fisik dari jaringan, meliputi kabel, routers, *switches* dan *network interface* cards (NIC). Susunan perangkat keras juga meliputi macam –macam protokol susunan perangkat keras; Ethernet adalah yang secara luas banyak digunakan. Ethernet bersandar pada konsep dari alamat *Media Access Control* (MAC), yang mana merupakan nilai 6-bit yang unik (seperti 00-02-B4-DA-92-2C ) yang secara permanen diberikan untuk *network interface card* khusus. Masing – masing frame berisi dua alamat MAC, yang mana menunjukkan alamat MAC dari NIC yang hanya diarahkan framenya dan alamat MAC setelah NIC yang framenya akan dikirim. Sebagai nilai frame yang melalui peralatan jaringan (seperti routers dan firewalls) pada jalannya antar host sumber yang asli dan *host* tujuan akhir, alamat MAC diperbaharui untuk menghubungi sumber dan tujuan lokal. Mungkin ada beberapa perangkat keras susunan transmisi yang terpisah dihubungkan secara bersama – sama dengan susunan transmisi IP tunggal.

Disamping alamat MAC, masing – masing bingkai juga berisi nilai EtherType, yang mana menunjukkan protokol frame yang berisi muatan yang menguntungkan (secara khususnya IP atau *Address Resolution Protocol* [ARP] ). Ketika IP digunakan, masing – masing alamat IP dipetakan ke alamat MAC tertentu. (berbagai alamat IP yang dapat dipetakan ke alamat MAC tunggal, jadi alamat MAC tidak secara khusus dibutuhkan mengidentifikasi untuk alamat IP)

### 6.1.5 Susunan penting dalam analisa data jaringan

Masing – masing dari empat susunan dari sederetan protokol TCP/IP berisi informasi penting. Susunan perangkat keras menyediakan informasi pada komponen fisik, ketika susunan lainnya menggambarkan aspek logis. Untuk kejadian tidak lebih dari jaringan, seorang analis dapat memetakan alamat IP (pengidentifikasi logis pada susunan IP) ke alamat

MAC dari *network interface card* tertentu (pengidentifikasi fisik susunan fisik) dengan cara demikian mengidentifikasi *host* menjadi menarik. Kombinasi dari nomor protokol IP (kolom susunan IP) dan nomor *port* (kolom – kolom susunan pengangkut) dapat memberitahukan seorang analis untuk aplikasi mana yang kebanyakan dapat dipastikan sedang digunakan atau ditargetkan. Hal ini dapat diverifikasi dengan menguji data susunan aplikasi.

Analisa data jaringan bersandar pada semua susunan. Ketika analis memulai utuk menguji data, mereka secara khusus memiliki informasi yang terbatas—kebanyakan seperti alamat IP yang menarik, dan mungkin informasi *port d*an protokol. Meski demikian, hal ini adalah informasi yang cukup untuk mendukung pencarian sumber data umum untuk informasi yang lebih banyak lagi. Dalam banyak kasus, susunan aplikasi berisi berisi aktifitas yang aktual yang menarik—kebanyakan serangan terhadap tempat yang mudah diserang dalam suatu aplikasi dan hampir semua penyalahgunaan meliputi penyalahgunaan aplikasi. Analis membutuhkan alamat IP sehingga mereka dapat mengidentifikasi host yang mungkin telah terlibat dalam aktifitas. Host mungkin juga berisi data tambahan yang harus digunakan dalam menganalisa suatu aktifitas. Walaupun banyak kejadian menarik yang mungkin saja tidak memiliki data level aplikasi yang relevan (seperti didistribusikannya penolakan dari serangan peralatan yang didesain untuk mengkonsumsi semua bandwidth jaringan), kebanyakan yang dilakukan; analisa data jaringan menyediakan dukungan penting untuk aktifitas analisa suatu aplikasi.

## 6.2 Sumber – sumber data jalur lalu lintas jaringan

Secara khusus organisasi – organisasi memiliki beberapa jenis sumber informasi pada jalur lalu lintas jaringan yang mungkin berguna. Sumber ini secara bersama – sama mengambil data penting dari keempat susunan TCP/IP. Bagian , perangkat lunak *security event management* dan *tool* analisa forensik jaringan-baik sebagai beberapa jenis lainnya dari sumber data. Untuk masing – masing yang seperti sumber, bagian ini menjelaskan tujuannya dan menggambarkan tipe data yang secara khusus dikumpulkan dan dapat berpotensi untuk dikumpulkan.

### 6.2.1 Firewalls dan Routers

Alat – alat dasar jaringan seperti *firewalls* dan *routers* dan alat dasar *host* seperti *firewalls* pribadi, memeriksa *network traffic* dan mengijinkan atau menolaknya didasarkan pada satu set peraturan. *Firewalls* dan *routers* biasanya dikonfigurasi ke informasi dasar *log*

untuk kebanyakan atau semua penolakan usaha berhubungan dan paket tanpa hubungan; banyak *log* pada setiap paket. Informasi di*load* secara khusus termasuk tanggal dan waktu paket yang diproses, alamat IP sumber dan tujuan dan protokol susunan pengangkut ( misalnya TCP, UDP, ICMP) dan informasi protokol dasar (misalnya nomor port TCP atau UDP, tipe dan kode ICMP). Isi dari paket pada umumnya tidak direkam.

*Firewalls* dan *routers* dasar jaringan yang melakukan penerjemahan alamat jaringan (NAT atau *Network Address Translation*) mungkin berisi data tambahan yang berharga berkenaan dengan jalur lalu lintas jaringan. NAT adalah proses memetakan alamat tersendiri dari jaringan internal ke satu atau lebih alamat publik pada jaringan yang dikoneksikan ke internet. NAT membedakan alamat internal *multiple* yang dipetakan ke alamat eksternal tunggal dengan memberikan sumber berbeda nomor port ke alamat eksternal untuk masing – masing alamat internal. Alat – alat NAT secara khusus mencatat alamat dan pemetaan *port* NAT.

Banyak *firewalls* juga bertindak sebagai *proxies. Proxy* menerima permintaan dari klien dan kemudian mengirim permintaan pada kepentingan klien untuk tujuan yang diinginkan. Ketika *proxy* digunakan, masing – masing koneksi yang sukses berusaha menghasilkan yang sebenarnya dalam kreasi dari koneksi yang terpisah : satu diantara klien dan server *proxy* dan antara server *proxy* lainnya juga tujuan benarnya. Server *proxy* mungkin informasi *log* dasar pada masing – masing koneksi. Banyak *proxy* merupakan aplikasi yang spesifik dan banyak yang sebenarnya melakukan banyak analisa dan validasi dari protokol aplikasi seperti HTTP. *Proxy* mungkin menolak permintaan klien yang kelihatannya tidak sah dan informasi log berkenaan dengan permintaan tersebut.

Disamping NAT dan *proxying services, firewalls* dan *router* mungkin juga melakukan fungsi lainnya, seperti pendeteksian gangguan dan *virtual private networking* (VPN). Fungsi ini didiskusikan dengan lebih detil, berturut - turut dalam bagian 6.2.3 dan 6.2.4.

### 6.2.2 Paket sniffer dan penganalisa protokol

Paket *sniffer* didesain untuk mengontrol jalur lalu lintas jaringan pada jaringan kabel atau tanpa kabel dan mengambil paket. Biasanya, *network interface card*(NIC) hanya menerima paket yang masuk dimana secara spesifik dimaksudkan untuk hal tersebut. Ketika NIC ditempatkan dalam *promiscuous mode,* hal itu menerima semua paket masuk yang dilihat, berkenaan dengan tujuan akhir yang dimaksudkan mereka. Paket *sniffer* secara umumnya bekerja dengan menempatkan NIC dalam *promiscuous mode;* kemudian user

mengkonfigurasi *sniffer* untuk mengambil semua paket atau hanya itu dengan karakteristik tertentu (misalnya, *port* TCP tertentu, sumber dan tujuan tertentu, alamat IP). Paket *sniffer* biasanya digunakan untuk mengambil jenis jalur lalu lintas tertentu untuk *troubleshooting* atau tujuan pemeriksaan. Sebagai contohnya, jika tanda IDS menunjukkan aktifitas jaringan yang tidak biasa diantara dua *host*, paket *sniffer* harus merekam semua paket diantara *host*, secara potensial menyediakan informasi tambahan untuk analis.

Banyak paket *sniffer* yang juga merupakan *protocol analyzers*, yang mana berarti mereka dapat memasang lagi *stream* dari paket sendiri dan mengartikan komunikasi yang menggunakan banyak dari ratusan atau ribuan protokol yang berbeda. *Protocol analyzers* biasanya dapat memproses tidak hanya jalur lalu lintas jaringan aktif, tetapi juga paket yang telah melakukan perekaman sebelumnya dalam pengambilan file dengan paket *sniffer*. *Protocol anlyzer* secara ekstrim berharga dalam menampilkan paket data mentah dalam format yang tidak dimengerti. *Protocol anlyzers* didiskusikan secara lebih dalam pada bagian 6.4 dan bagian 7.

### 6.2.3 Sistem pendeteksi gangguan

*Network-based intrusion detection systems* (IDS) menyelenggarakan paket *sniffing* dan *analyze network traffic* untuk mengidentifikasi aktifitas yang mencurigakan dan catatan yang relevan dengan informasi. Karakteristik kontrol *Host based IDS* dari sistem dan menjadi kejadian dalam sistem, yang mana dapat meliputi jalur lalu lintas jaringan. Tidak seperti dasar jaringan sensor IDS, yang mana dapat memonitor semua jalur lalu lintas jaringan pada segmen jaringan tertentu, perangkat lunak *host-based* IDS yang diharapkan untuk hanya memonitor jalur lalu lintas jaringan untuk *host* pada tempat diinstalnya perangkat lunak tersebut. Untuk masing – masing peristiwa yang mencurigakan, perangkat lunak IDS secara khusus mencatat karakteristik kejadian dengan dasar sama yang *firewalls* dan *router* catat (misalnya, tanggal dan waktu, sumber dan tujuan alamat IP, protokol, karakteristik protokol dasar) yang dikenal sebagai informasi aplikasi yang spesifik (misalnya nama user, nama file, perintah kode status). Perangkat IDS juga mencatat informasi yang menunjuk pada aktifitas yang mungkin dimaksud. Contohnya meliputi jenis – jenis dari serangan (misalnya *buffer overflow*), yang mudah diserang ditargetkan, serangan yang kelihatan sukses atau gagal dan penunjuk untuk informasi yang lebih banyak pada serangan.

Banyak IDS yang dapat dikonfigurasi untuk mengambil paket yang berhubungan dengan aktifitas yang mencurigakan. Hal ini dapat dimulai dari merekam hanya pada paket

yang menggerakkan IDS ke label aktifitas yang mencurigakan, untuk merekam sesi istirahatnya. Banyak IDS genap memiliki kemampuan untuk menyimpan semua sesi untuk waktu dengan periode yang pendek, jadi jika hal yang mencurigakan tersebut terdeteksi, aktifitas sebelumnya dalam sesi yang sama dapat dikarantina. Paket utama yang diambil merupakan deteksi yang mencurigakan, analis dapat meninjau ulang hal itu ketika memvalidasi sinyal IDS dan menginvestigasi aktifitas yang mencurigakan. Banyak juga IDS yang memiliki kemampuan *intrusion prevention*, yang mana berarti mereka secara aktif berusaha untuk menghentikan serangan dalam perkembangannya. Penggunaan lainnya dari fitur *intrusion prevention* harus ditunjukkan dalam *log* IDS.

### 6.2.4 Remote Access

*Remote access servers* adalah alat seperti VPN *gateways* dan *modem servers* yang memfasilitasi koneksi diantara jaringan. Hal ini selalu meliputi sistem eksternal yang dihubungkan ke sistem internal melalui *remote access server*, tetapi harus juga termasuk sistem internal yang dihubungkan sistem ke eksternal atau internal . *Remote access servers* secara khusus mencatat asal mula masing – masing hubungan dan mungkin juga menunjukkan laporan *user* mana yang asli untuk masing – masing sesi. Jika *remote access server* memberikan sebuah alamat IP untuk *user* jauh, hal ini juga seperti untuk dicatakan. Banyak *remote access server* juga menyediakan fungsi menyaring paket; hal ini secara khusus meliputi pencatatan yang mirip untuk hal itu dimana untuk *firewalls* dan *router*, yang diuraikan dalam bagian 6.2.1. *Remote access servers* secara khusus bekerja pada level jaringan, mendukung penggunaan banyak aplikasi yang berbeda. Karena server tidak mengerti fungsi dari aplikasi, mereka biasanya tidak merekam banyak data dari aplikasi yang spesifik.

Sebagai tambahan untuk *remote access servers*, secara khusus organisasi menggunakan banyak aplikasi yang khususnya didesain untuk menyediakan *remote access* ke sistem operasi *host* tertentu. Contohnya meliputi penjaminan *shell*(SSH), *telnet,* server terminal dan perangkat lunak remote control. Seperti aplikasi yang dapat secara khusus dikonfigurasi ke informasi dasar *log* untuk masing – masing koneksi, meliputi alamat sumber IP dan laporan *user*.. Organisasi juga secara khusus menggunakan banyak aplikasi yang sedikit diakses, seperti aplikasi klien/server. Banyak dari aplikasi tersebut yang mungkin juga informasi dasar *log* untuk koneksi.

Walaupun kebanyakan *remote access* dihubungkan dengan pencatatan yang terjadi pada *remote access server* atau aplikasi server, dalam banyak kasus seorang klien juga mencatat informasi yang dihubungkan untuk suatu hubungan.

### 6.2.5 Perangkat lunak Security event management (manajemen peristiwa keamanan)

Perangkat lunak *Security Event Management* (SEM) sanggup untuk mengirim informasi kejadian keamanan dari berbagai jalur lalu lintas jaringan yang dihubungkan sumber data kejadian keamanan (misalnya *log* IDS, *log firewall*) dan menghubungkan kejadian diantara sumber – sumber. Hal ini biasanya bekerja dengan mengirim salinan *log* dari berbagai sumber data diluar saluran yang aman, menormalisasi *log* kedalam format standar, kemudian mengidentifikasi kejadian yang dihubungkan dengan menyamakan alamat IP, *timestamps* dan karakteristik lainnya. Produk SEM biasanya tidak menghasilkan data kejadian yang asli; malahan, mereka menghasilkan *metaevents* yang berdasar pada data kejadian yang didatangkan. Banyak produk SEM tidak hanya dapat mengidentifikasi aktifitas kejahatan, seperti serangan dan infeksi dari virus tetapi mereka juga dapt mendeteksi penyalahgunaan dan dalam penggunaan sistem dan jaringan yang tidak sesuai. Perangkat lunak SEM dapat berguna dalam membuat banyak sumber dari informasi jalur lalu lintas jaringan bersifat dapat diakses melalui single interface.

Karena produk SEM dapat menangani hampir apapun sumber data kejadian keamanan, seperti *log* OS, sinyal perangkat lunak anti virus dan alat *log* keamanan secara fisik, SEM produk mungkin berisi berbagai informasi yang luas berkenaan dengan suatu kejadian. Bagaimanapun, hal tersebut secara khusus hanya untuk banyak data dasar yang menjadi sisa yang diambil, sebagai contohnya, jika ada sebuah paket catatan IDS, paket tersebut tidak mungkin dapat dipindahkan ke SEM karena keterbatasan *bandwidth* dan penyimpanan. Juga karena kebanyakan sumber data merekam informasi dalam format berbeda, produk SEM secara khusus menormalisasi data—mengubah masing – masing dasar data untuk format standar dan melabelkan data secara terus menerus. Meskipun hal ini menguntungkan untuk analisa (diuraikan dalam bagian 6.4), proses penormalisasi mungkin adakalanya memperkenalkan kerusakan pada data atau kenapa banyak data dapat hilang. Untungnya, produk SEM secara khusus tidak mengubah sumber data asli,jadi analis dapat memverifikasi keakuratan data jika dibutuhkan.

### 6.2.6 Alat analisa forensik jaringan

Alat analisa forensik jaringan (NFAT) secara khusus menyediakan kesamaan fungsional sebagai paket *sniffer*, penganalisa protokol dan perangkat SEM dalam produk tunggal. Ketika perangkat lunak SEM dipusatkan pada penghubungan kejadian diantara pengadaan sumber data (yang mana secara khusus meliputi banyak sumber jalur lalu lintas jaringan yang direlasikan), perangkat NFAT adalah yang terutama difokuskan pada pengumpulan dan menganalisa jalur lalu lintas jaringan. Perangkat lunak NFAT juga menawarkan fitur tambahan yang lebih lanjut memfasilitasi analisa data jaringan, seperti yang berikut ini :

1. Merekontruksi kejadian dengan mengulangi menggunakan jalur lalu lintas jaringan dalam *tool*, menyusun dari sesi individu (misalnya mengirim pesan secara instan diantara dua *user*) ke semua sesi selama periode waktu tertentu. Kecepatan mengulangi penggunaan dapat secara khusus menjadi diatur sessuai kebutuhan.
2. Memvisualisasi alur jalur lalu lintas dan hubungan antara *host*. Banyak *tool* dapat tetap berlandaskan alamat IP, nama domain atau data lainnya ke lokasi fisikal dan memproduksi peta geografi dari aktifitas.
3. Membangun profil pada aktifitas yang khusus dan mengidentifikasi penyimpangan yang penting.
4. Mencari isi aplikasi untuk kata kunci (misalnya “ confidential”,” proprietary”)

### 6.2.7 Sumber lainnya

Banyak organisasi memiliki sumber lainnya pada informasi network traffic yang mungkin dapat digunakan sebagai analisa dalam banyak kapasitas, meliputi yang dibawah ini:

1. *Dynamic Host Configuration Protocol (DHCP) servers.*

   Layanan DHCP memberikan alamat IP untuk *host* pada jaringan yang dibutuhkan. Banyak *host* berkemampuan memiliki alamat IP yang statis, yang mana berarti dimana mereka selalu menerima penempatan alamat IP yang sama; bagaimanapun, kebanyakan *host* secara khusus menerima tugas penempatan. Hal ini berarti dimana *host* dibutuhkan untuk memperbaharui secara tetap penempatan alamat IP mereka dan dimana tidak ada jaminan untuk mereka yang akan ditempatkan pada alamat yang sama. Server DHCP mungkin berisi *log* penempatan yang meliputi alamat MAC dan waktu penetapan yang terjadi.

2. Perangkat Lunak untuk memonitoring jaringan

   Perangkat Lunak untuk memonitoring jaringan didesain untuk mengamati jalur lalu lintas jaringan dan mengumpulkan statistiknya. Sebagai contohnya, hal itu mungkin mencatat informasi tingkat tinggi pada alur jalur lalu lintas untuk segmen jaringan tertentu, seperti jumlah dari *bandwidth* yang secara khusus digunakan oleh berbagai macam protokol. Perangkat Lunak untuk memonitoring jaringan mungkin juga mengkoleksi informasi lebih detil pada aktifitas jaringan, seperti ukuran alat – alat dan sumber dan tujuan alamat IP dan port untuk masing – masing paket. Banyak switch yang diatur dan alat jaringan lainnya menawarkan kemampuan memonitor jaringan dasar, seperti mengumpulkan statistik pada penggunaan bandwidth.

3. Catatan penyedia layanan internet.

   Penyedia layanan internet (*Internet service providers* /ISP) mungkin mengkoleksi data jalur lalu lintas jaringan yang dihubungkan sebagai bagian operasi normalnya dan ketika menginvestigasi aktifitas yang tidak biasa, seperti volume tinggi yang luar biasa pada jalur lalu lintas atau serangan yang terlihat. Catatan normal ISP seringkali berkemampuan hanya menjaga untuk beberapa hari atau beberapa jam. Bagian 6.3.1 mendiskusikan pertimbangan legal dengan memperoleh data jalur lalu lintas jaringan dari ISP dan bagian ketiga lainnya.

4. Aplikasi *client/server*

   Banyak aplikasi *client/server* yang digunakan diluar jaringan yang mungkin mencatat informasi berkenaan dengan usaha berhasil atau tidaknya penggunaan, yang mana harus meliputi koneksi yang dihubungkan seperti alamat IP *client* dan *port*. Kolom data dicatat (jika ada) merubah secara luas diantara aplikasi – aplikasi.

5. Konfigurasi jaringan *host* dan koneksi.

   Bagian 5.1.2 dan 5.2.1 menguraikan jenis – jenis dari informasi jaringan yang dapat diperoleh dari *host* secara individu, termasuk *port* TCP dan UDP ditempat *host* mendengarkan.

## 6.3 Memperoleh data jalur lalu lintas jaringan

Sebagaimana yang diuraikan pada bagian 6.2, secara khusus organisasi memiliki data jaur jaringan yang dicatat dalam banyak tempat selama operasi normal. Organisasi juga menggunakan mekanisme rekaman data yang sama untuk memperoleh data tambahan pada

dasar yang dibutuhkan ketika menginvestigasi kejadian atau menyelesaikan masalah. Untuk contohnya, seorang penanggung jawab jaringan atau orang yang menangani kejadian bekemampuan menyebarkan paket sniffer untuk menguji paket yang tidak biasa yang dikirim oleh host.

Data jalur lalu lintas jaringan biasany dicatat ke log atau disimpan dalam paket file pengambil. Dalam kebanyakan kasus , memperoleh data itu simpel seperti memperoleh data salinan dari log dan paket file pengambil. Bagian 4 menguraikan bagaimana untuk mendapatkan file. Jika data tidak disimpan di file (misalnya peta alur jalur lalu lintas diperlihatkan secara grafik, data diperlihatkan hanya pada console screen), pengambilan layar atau gambar pada layar mungkin dibutuhkan. Meskipun memperoleh data jalur lalu lintas jaringan secara khususnya secara jujur, beberapa data penting tersebut legal dan persoalan secara teknik dapat membuat perolehan data tersebut lebih sulit.

### 6.3.1 Pertimbangan Legal

memperoleh jalur lalu lintas jaringan dapat merupakan banyak persoalan legal, seperti pengambilan (ketidaksengajaan atau kebetulan ) informasi dengan kebebasan atau implikasi keamanan, seperti password atau isi dari email. Hal ini harus membuka informasi ke anggota staf yang menganalisa data atau melaksanakn sistem perekaman (misalnya sensor IDS). Organisasi harus memiliki kebijaksanaan dalam penempatan berkenaan dengan penanganan penyingkapan informasi sensitif yang kurang hati - hati . masalah lainnya dengan mengambil data seperti email dan dokumen berupa teks yang merupakan penyimpanan jangka panjang dari hal seperti informasi yang mungkin melanggar kebijaksanaan hak tetap data organisasi. Hal tersebut juga penting untuk memiliki kebijaksanaan berkenaan dengan memonitor jaringan, seperti halnya sinyal pada sistem yang menunjukkan aktifitas yang mungkin dimonitor.

Meskipun kebanyakan perolehan data jalur lalu lintas jaringan terjadi sebagai bagian operasi reguler, hal itu mungkin juga diperoleh sebagai bagian dari menyelesaikan masalah atau menangani kejadian. Dalam kasus yang kemudian, hal itu penting untuk mengikuti proses yang konsisten dan dokumen semua aksi yang dilakukan. Untuk contoh, mencatat semua paket yang dikirim dan diterima oleh user tertentu seharusnya dilakukan hanya setelah permintaan formal dan proses persetujuan yang secara utuh berhasil. Organisasi harus memiliki kebijaksanaan yang jelas mengutarakan jenis memonitor yang seperti apa yang

dapat atau tidak dilakukan tanpa persetujuan dan menguraikan atau mereferensikan prosedur permintaan detil dan proses persetujuan.

Kebebasan pribadi dapat menjadi perhatian yang lebih besar ke organisai,banyak yang sudah menjadi kurang berkeinginan untuk berbagi informasi satu dengan lainnya, meliputi data jaringan. Sebagai contoh, kebanyak ISP sekarang memerlukan perintah dari luar sebelum menyediakan informasi apapun berhubungan degnan aktifitas jaringan yang mencurigakan yang mungkin telah melintas melalui infrastruktur mereka. Meskipun hal ini menjaga kebebasan pribadi dan mengurangi kewajiban dan beban dari ISP, hal itu juga melambatkan proses investigasi. Hal ini terutama sekali menantang ketika organisasi mengusahakan untuk melacak suatu serangan network-based yang berkelanjutan kesumbernya, terutama jika jalur lalu lintas lewat melalui beberapa ISP.

### 6.3.2 Persoalan secara teknis

Ada beberapa persoalan teknis berpotensi yang mungkin menghalangi perolehan data pada jalur lalu lintas jaringan.

Bagian ini menguraikan beberapa persoalan utama dan menyediakan bimbingan terhadap apa, meskipun terdapat perbedaan, dapat dilakukan untuk mengurangi masing – masing persoalan.

1. Tempat penyimpanan data.

   Ketika volume yang besar pada aktifitas jaringan terjadi, terutama sekali selama kejadian yang kurang baik seperti serangan, *log* mungkin mencatat banyak kejadian dalam waktu yang singkat. Jika tempat penyimpanan yang tidak cukup tersedia, informasi tentang aktifitas terbaru mungkin di*overwrite* dan hilang. Organisasi harus memperkirakan jenis dan penggunaan log maksimum, menentukan bagaimana banyak jam atau hari yang berharga dari data untuk dikuasai dan memastikan sistem dan aplikasi tersebut memiliki cukup tempat penyimpanan yang tersedia untuk menemukan tujuan itu.

2. Jalur lalu lintas yang dienkripsi.

   Ketika protokol seperti IPsec, SSH dan SSL digunakan untuk mengenkripsi jalur lalu lintas jaringan, alat yang memonitor jalur lalu lintas jaringan sepanjang alur yang dienkripsi dapat melihat hal – hal karakteristik mendasar pada jalur lalu lintas saja, seperti alamat IP dari sumber dan tujuan. Jika VPN atau teknik "*tunneling*" digunakan, alamat IP dapat dijadikan untuk "*tunnel*"nya sendiri dan bukan sumber dan tujuan sebenarnya dari aktifitas. Untuk perolehan data pada pendekripsian jalur lalu lintas, sumber data diperlukan untuk diposisikan ditempat mana aktifitas yang didekripsikan dapat dilihat.

Sebgai contoh, penempatan sensor IDS dengan segera dibelakang VPN *gateway* dapat menjadi efektif pada saat mengidentifikasi aktifitas ganjil dalam komunikasi yang didekripsikan. Jika komunikasi dienkripsi semua cara untuk ke *host* internal (misalnya *SSL-encrypted* sesi Web) maka alat – alat yang memonitor jalur lalu lintas jaringan tidak dapat melihat paket yang dideskripsikan

3. Menjalankan layanan pada port yang tak terduga

   Aplikasi seperti sistem pendeteksi gangguan dan penganalisa protokol seringkali bersandar pada nomor *port* untuk mengidentifikasi layanan mana yang digunakan untuk memberikan koneksi. Sayang sekali, seperti yang diuraikan dalam bagian 6.1.2, kebanyakan layanan dapat dijalankan pada nomor *port* manapun. Jalur lalu lintas menyertakan layanan yang dijalankan pada nomor *port* tak terduga yang tidak dapat diambil, dimonitor atau dianalisa dengan baik, menyebabkan penggunaan layanan yang tidak sah (misalnya menyediakan layananWeb pada port tidak lazim) tidak akan terdeteksi. Motivasi lainnya adalah mendahului jalur lalu lintas melalui alat perimeter yang menyaring berdasarkan nomor *port*. Ada beberapa cara untuk berusaha mengidentifikasi penggunaan *port* yang tidak sah, mencakup hal – hal berikut ini:

   1. Mengkonfigurasi sensor IDS untuk siaga pada koneksi yang melibatkan *port* server yang tidak diketahui.
   2. Mengkonfigurasi *proxy* aplikasi atau sensor IDS yang menyelenggarakan analisa protokol untuk siaga pada koneksi yang menggunakan protokol yang tak diduga (misalnya jalur lalu lintas FTP menggunakan *port* HTTP standar)
   3. Menyelenggarakan "*monitoring*" alur *network traffic* dan mengidentifikasi alur *network traffic* baru dan yang tidak biasa.
   4. Mengkonfigurasi penganalisa protokol untuk menganalisa arus tertentu sebagai hal lain

4. Pengganti Tujuan akses

   Penyerang seringkali memasuki jaringan dari pengganti tujuan akses untuk menghindari pendeteksian oleh kontrol keamanan yang memonitor tujuan akses utama, seperti *internet gateway* organisasi. Sebuah contoh klasik dari pengganti tujuan akses adalah *modem* dalam *workstation user*. Jika penyerang dapat menyambungkan ke *workstation* dan mendapatkan akses, maka serangan dapat menjadi diluncurkan dari *workstation* melawan host lainnya. Dalam kasus yang demikian, sedikit atau tidak adanya informasi berkenaan

dengan aktifitas jaringan mungkin dapat dicatatkan karena aktifitas tersebut tidak menerobos *firewalls,* IDS dimonitor segmen jaringan dan tujuan pengumpulan data umum lainnya. Organisasi secara khusus menunjuk ini dengan membatasi penggantian tujuan akses, seperti modem dan tujuan akses tanpa kabel dan memastikan bahwa masing – masing hal tersebut dimonitor dan terbatas melalui *firewalls*, sensor IDS dan kontrol lainnya

5. Kegagalan memonitor.

   Hal yang tak bisa diacuhkan, sistem dan aplikasi dapat mengalami kegagalan atau kadangkalanya diluar jalur lalu lintas untuk berbagai pertimbangan (misalnya pemeliharaan sistem, kegagalan perangkat lunak, serangan). Dalam kasus yang dipersembahkan sistem yang memonitor, seperti sensor IDS, penggunaan peralatan yang berlebihan (misalnya dua sensor untuk sktifitas yang sama ) dapat mengurangi dampak dari kegagalan memonitor. Strategi lainnya akan dilakukan kegiatan memonitor dalam berbagai tingkatan, seperti mengkonfigurasi dasar jaringan dan *hostbased firewalls* untuk mencatatkan koneksi.

## 6.4 Menguji data jalur lalu lintas jaringan

Ketika kejadian yang menarik telah dapat diidentifikasikan, analis mengekstrak dan meneliti data *network traffic* dengan tujuan menentukan apa yang telah terjadi dan bagaimana sistem organisasi dan jaringan telah dipengaruhi. Hal ini mungkin sesederhana meninjau ulang sedikit dari masukkan *log* pada sumber data tunggal dan menentukan bahwa suatu kejadian merupakan tanda bahaya palsu atau sebagai hal kompleks lainnya yang secara sekuen meninjau ulang dan meneliti kumpulan data untuk menentukan arti dan tujuan yang mungkin dari kejadian. Meskipun demikian, bahkan untuk kasus sederhana yang secara relatif memvalidasi sedikit dari masukkan log dapat menjadi keanehan yang dilibatkan dan memakan banyak waktu.

Meskipun *tool* sekarang (misalnya perangkat lunak SEM, perangkat lunak NFAT) dapat berguna dalam mengumpulkan dan mempresentasikan data jalur lalu lintas jaringan, *tool* seperti itu memiliki kemampuan anlisa yang agak terbatas dan hanya dapat digunakan secara efektif dengan analis yang terlatih dengan baik; berpengalaman. Sebagai tambahan untuk mengerti *tool* , analis juga perlu untuk memiliki pengetahuan yang mendalam tentang prinsip jaringan, jaringan umum dan protokol aplikasi, jaringan dan produk keamanan aplikasi

dan ancaman *network-based* dan serangan. Hal ini juga penting untuk memiliki pengetahuan yang baik tentang lingkungan organisasi, seperti arsitektur jaringan dan alamat IP yang digunakan oleh aset kritis (misalnya *firewalls, server* yang dapat diakses secara umum), seperti halnya informasi yang mendukung aplikasi dan sistem operasi yang digunakan oleh organisasi. Jika analis mengerti organisasi yang normal menghitung basis, seperti khususnya pola penggunaan sistem dan jaringan ke perusahaan lainnya, pekerjaan mereka harus lebih mudah dan cepat untuk dilakukan. Analis harus juga memiliki pemahaman tentang perusahaan untuk masing – masing sumber data *network traffic*, seperti halnya akses yang mendukung material, seperti dokumentasi tanda terdeteksinya gangguan. Analis perlu untuk mengerti nilai karakteristik dan relatif dari masing – masing sumber data sedemikian sehingga mereka dapat melokasikan data yang relevan secepatnya.

Sebab proses analisis sering kompleks dan analis perlu pengetahuan tentang jaringan yang luas dan beberapa area keamanan informasi untuk meneliti data jalur lalu lintas jaringan secara efektif dan mengembangkan kesimpulan, hal ini diluar lingkup dari pemandu ini untuk menguraikan teknik meneliti data dan menggambarkan kesimpulan dalam situasi yang kompleks. Maka, bagian ini fokus pada langkah dasar dari proses pengujian dan juga menyoroti banyak persoalan teknis yang penting dipertimbangkan oleh analis.

### 6.4.1 Mengidentifikasi suatu peristiwa yang penting

Langkah pertama dalam proses pengujian dalah mengidentifikasi suatu peristiwa yang menarik. Secara khusus ini terjadi melalui satu dari dua metode, sebagai berikut :

1. Seseorang didalam organisasi tersebut (misalnya *help desk agent, system administrator, security administrator* ) telah diterima suatu indikasi, seperti tanda yang diotomatiskan atau komplain dari *user,* yang mungkin hal itu berupa keamanan dan persoalan yang terkait dengan operasional. Seorang analis harus dapat diminta untuk riset aktifitas yang bersesuaian.
2. Tugas regular analis meliputi meninjau ulang data peristiwa keamanan (misalnya memonitor IDS, memonitor jaringan, menijau ulang catatan *firewalls*). Selama meninjau ulang, analis mengidentifikasi peritiwa yang penting dan menentukan bahwa hal tersebut harus diteliti lebih lanjut.

Ketika peristiwa yang menarik telah diidentifikasi, analis harus mengetahui beberapa informasi dasar mengenai peristiwa sebagai basis dari riset. Dalam kebanyakan kasus,

peristiwa yang dideteksi melalui sumber data *network traffic*, seperti sensor IDS atau *firewall*, sehingga analis dapat dengan mudah ditunjuk sumber data itu untuk lebih banyak lagi informasi. Bagaimanapun, dalam banyak kasus, seperti komplain *user*, hal itu mungkin tidak nyata yang mana sumber data (jika ada) mungkin berisi informasi relevan yang mana *host* atau jaringan mungkin dilibatkan. Analis mungkin perlu untuk bersandar pada banyak informasi umum, seperti laporan beberapa sistem dasar keempat telah me"*reboot"* dirinya mereka sendiri. Meskipun pengujian data begitu mudah jika informasi kejadian spesifik (misalnya alamat IP dari sistem yang dipengaruhi), bahkan informasi umum menyediakan kepada analis dengan tujuan awal untuk menemukan sumber data yang relevan.

### 6.4.2 Menguji sumber data

Sebagaimana yang diuraikan dalam bagian 6.2, organisasi mungkin memiliki banyak sumber jaringan data terkait dengan *network traffic*. Kejadian tunggal yang menarik dapat dicatat banyak sumber data, tetapi hal itu tidak efisien atau tidak praktis untuk memeriksa masing – masing sumber secara individu. Untuk pengujian data awal kejadian, analis secara khusus bersandar pada sedikit sumber data primer, seperti *IDS console* yang menampilkan tanda dari sensor IDS atau perangkat SEM atau NFAT yang memperkuat banyak sumber data lainnya dan mengorganisir data tersebut. Hal ini tidak hanya solusi yang efesien, tetapi juga dalam kebanyakan kasus kejadian yang menarik yang diidentifikasi oleh tanda dari salah satu sumber data primer. Maka analis terlebih dulu secara khusus berkonsultasi satu atau sedikit sumber data primer.

Untuk masing – masing sumber data yang diuji, analis harus mempertimbangkan ketepatannya. Secara umum, harus memiliki banyak kepercayaan dalam sumber data asli dibanding sumber data yang menerima data yang dinormalisasi (dimodifikasi) dari sumber lainnya. Juga analis harus memvalidasi data apapun yang berdasarkan pada meneliti dan menginterpretasikan data, seperti IDS dan tanda SEM. Tidak ada *tool* untuk mengidentifikasi aktifitas kejahatan yang secara komplit akurat; mereka menghasilkan keduanya yaitu nilai positif yang salah (salah melaporkan aktifitas tidak berbahaya sebagai kejahatan) dan nilai negatif yang salah (salah mengklasifikasikan aktifitas kejahatan sebagai hal yang tidak berbahaya). *Tools* seperti NFAT dan IDS mungkin juga menghasilkan tanda yang tidak akurat jika mereka tidak memproses semua proses didalam suatu koneksi. Memvalidasi harus didasarkan pada pengujian data tambahan (misalnya paket mentah, informasi pendukung yang diambil oleh sumber lainnya), meninjau ulang ketersediaan informasi pada validitas tanda

tanda (misalnya komentar vendor pada nilai positif salah yang dikenal) dan pengalaman lalu dengan tool yang dipermasalahkan. Dalam kebanyakan kasus, seorang analis berpengalaman dapat secara cepat menguji data pendukung dan menentukan bahwa suatu tanda tanda bernilai positif salah dan tidak membutuhkan penyelidikan lebih lanjut.

Analis mungkin juga perlu untuk menguji sumber data jalur lalu lintas jaringan sekunder, seperti *host-based firewall logs* dan *packet capture* dan sumber yang data jalur lalu lintas jaringan, seperti *host* OS yang mengaudit *logs* dan *logs* perangkat lunak antivirus. Alasan yang paling umum untuk melakukan ini adalah sebagai berikut :

1. Tidak ada data pada sumber utama. Dalam beberapa hal, jalur lalu lintas jaringan sumber data secara khusus tidak berisi bukti aktifitas tersebut. Untuk contohnya, suatu serangan mungkin telah terjadi diantara dua *host* pada segmen jaringan internal yang tidak dimonitor atau dikontrol oleh alat keamanan jaringan. Analis perlu kemudian mengidentifikasi sumber data lain yang mungkin dan mengujinya untuk bukti.
2. Data yang tidak divalidasi atau tidak cukup pada sumber utama. Analis mungkin perlu untuk menguji sumber data sekunder jika sumber data utamanya tidak berisi informasi yang cukup atau analis perlu untuk memvalidasi data. Setelah meninjau ulang satu atau lebih sumber data utama, analis kemudian meng-*query* sumber data sekunder yang sesuai berdasarkan pada data yang bersangkutan dari sumber data utama. Sebagai contohnya, jika catatan IDS menunjukkan serangan terhadap sistem pada alamat IP 10.20.30.40 dengan alamat sebenarnya 10.3.0.1, kemudian meng-*query* sumber data lainnya dengan menggunakan satu atau kedua alamat IP yang mungkin menemukan data tambahan mengenai aktifitas tersebut. Analis juga menggunakan *timestamps*, protokol, *port number* dan *data field* umum lainnya untuk membatasi pencarian sebagaimana dibutuhkan.
3. Sumber data terbaik ditempat lain. Adakalanya, sumber terbaik dari data jalur lalu lintas jaringan mungkin ditempatkan pada *host* tertentu, seperti *host-based* f*irewall* dan *log* IDS pada sistem yang telah diserang. Meskipun sumber data seperti itu dapat sangat berguna, data mereka mungkin telah diubah atau dihilangkan selama suksesnya serangan tersebut.

Jika data tambahan diperlukan tetapi tidak bisa ditempatkan dan aktifitas yang mencurigakan masih tetap terjadi, analis mungkin perlu untuk melakukan lebih banyak lagi aktifitas untuk memperoleh data. Sebagai contohnya, seorang analis dapat melakukan pengambilan paket di titik – titik yang sesuai pada jaringan untuk mengumpulkan lebih

banyak informasi. Cara lain untuk mengumpulkan banyak informasi meliputi mengkonfigurasi *firewalls* atau *routers* untuk mencatat banyak informasi pada tertentu, menentukan *IDS signature* ke pengambilan paket untuk aktifitas dan menulis *custom IDS signature* yang siaga ketika terjadi aktifitas spesifik. Lihat bagian 6.2 untuk petunjuk tambahan pada *tool y*ang dapat memperoleh data. Mendapatkan data tambahan mungkin dapat berguna jika aktifitas berkelanjutan ; jika aktifitas telah diakhiri, tidak ada kesempatan untuk mendapatkan data tambahan.

#### 6.4.2.1 Nilai sumber data

Seperti yang diuraikan dalam bagian 6.2, organisasi secara khusus memiliki banyak sumber berbeda dari data *network traffic*. Karena informasi direkam oleh masing – masing sumber yang dapat saling tukar secara luas, nilai dari masing – masing sumber mungkin dapat saling tukar secara umum dan untuk kasus yang spesifik. Materi berikut menguraikan nilai khusus dari sumber data umum kebanyakan dalam analisis data jaringan :

1. Perangkat lunak IDS. Data IDS sering memulai tujuan untuk menguji aktifitas yang mencurigakan. Tidak hanya melakukan usaha khusus IDS untuk mengidentifikasi kejahatan *network traffic* pada semua lapisan TCP/IP, tetapi mereka juga mencatat banyak data field (dan kadang – kadang paket mentah) yang dapat berguna dalam kejadian memvalidasi dan mengkorelasikan mereka dengan sumber data lainnya. Seperti yang telah diuraikan diawal, perangkat lunak IDS memproduksi *false positives*, jadi tanda IDS perlu untuk divalidasi. Tingkat yang mana untuk hal ini dapat dilakukan tergantung pada jumlah data yang direkam berhubungan dengan tanda dan informasi yang disediakan untuk analis pada karakteristik *signature* atau metode mendeteksi anomali yang digerakkan oleh tanda tersebut.
2. SEM Perangkat lunak. Idealnya, SEM dapat sangat bermanfaat untuk analisis data karena hal itu dapat secara otomatis mengkorelasikan kejadian antara beberapa sumber data, kemudian mengekstrak informasi yang relevan dan diberikan kepada user. Bagaimanapun, karena fungsi perangkat lunak SEM dengan menghasilkan data dari banyak sumber lainnya, nilai SEM bergantung atas sumber data yang dipelihara didalamnya, bagaimana percaya kepada masing – masing sumber data dan bagaimana baiknya perngakat lunak dapat menormalisasikan data dan menghubungkan peristiwa.

3. Perangkat lunak. NFAT . Perangkat lunak NFAT ini didesain secara khusus untuk membantu dalam menganalisis jalur lalu lintas jaringan, jadi hal tersebut merupakan hal yang secara khusus berharga untuk ditinjau ulang kejadiannya ketika telah dimonitor. Perangkat lunak NFAT pada umumnya menawarkan fitur yang mendukung analisis, seperti rekonstruksi dan visualisasi jalur; bagian 6.2.6 menguraikan hal ini dengan lebih dalam.
4. *Firewalls, Routers, Proxy Servers*, and *Remote Access Servers*. Dengan sendirinya, data dari sumber ini kadang – kadang merupakan nilai yang kecil. Menguji data dari waktu ke waktu mungkin menunjukkan kecenderungan keseluruhan, seperti suatu peningkatan dalam usaha yang dihalangi koneksi. Bagaimanapun, karena sumber ini secara khusus mencatat sedikit informasi tentang masing – masing kejadian, data menyediakan pengertian kecil yang mendalam menyangkut kejadian. Juga, mungkin ada banyak kejadian yang dicatat tiap harinya, sehingga data – data kecil dapat berlimpah. Nilai utama dari data adalah untuk menghubungkan kejadian yang direkam oleh sumber lainnya. Sebagai contoh, jika suatu host disetujui dan sensor jaringan IDS dideteksi suatu serangan, menyangsikan *firewalls log* untuk melibatkan kejadian penyerangan alamat IP sebenarnya mungkin telah diusahakan untuk dikompromikan. Pemetaan alamat (misalnya NAT) yang dilakukan oleh alat ini sangat penting untuk analisis data jaringan karena alamat IP yang jelas seorang *attacker* atau korban mungkin benar – benar dapat digunakan oleh ratusan atau ribuan host. Secara kebetulan, analis kadang – kadang meninjau ulang log untuk menentukan alamat internal mana yang telah digunakan.
5. Server DHCP. Server DHCP secara khusus dapat dikonfigusrai ke log masing – masing penempatan alamat IP dan dihubungkannya alamat MAC. Bersama *timestamp*. Informasi ini sapat berguna untuk analis dalam aktifitas yang dilakukan *host* dengan menggunakan alamat IP tertentu. Bagaimanapun analis harus ingat akan kemungkinan bahwa *attacker* pada jaringan internal organisasi sudah memalsukan alamat MAC mereka atau alamat IPnya, yang mana dikenal sebagai *spoofing*.
6. Paket *Sniffers*. Dari semua sumber data jalur lalu lintas jaringan, paket *sniffer* dapat mengumpulkan kebanyakan informasi pada aktifitas jaringan. Meskipun, *sniffer* mungkin mengambil volume yang besar dari data yang tidak berbahaya sebagai jutaan atau miliaran paket dan secara khusus mereka tidak menyediakan indikasi paket mana yang mungkin berisi aktifitas kejahatan. Dalam kebanyakan kasus, paket *sniffer* merupakan hal terbaik

yang digunakan untuk menyediakan banyak data pada suatu kejadian bahkan alat lainnya atau perangkat lunak yang telah diidentifikasi sebagai kejahatan yang mungkin. Banyak catatan organisasi pada umumnya atau semua paket untuk beberapa periode waktu sedemikian sehingga ketika kejadian terjadi, data jaringan yang mentah tersedia untuk analisis. Data paket *sniffer* ditinjau ulang dengan baik oleh *protocol analyzer*, yang mana menginterpretasikan data untuk analis berdasar pada pengetahuan tentang standar protokol dan implementasi umum.

7. Memonitor jaringan. Perangkat lunak untuk memonitor jaringan berguna dalam mengidentifikasi penyimpangan yang penting dalam alur *network traffic* normal, hal yang seperti itu disebabkan oleh *distributed denial of service* (DDoS) *attacks*. Selama serangan ini, ratusan atau ribuan sistem meluncurkan serangan secara simultan melawan *host – host* tertentu atau jaringan. Perangkat lunak untuk memonitor jaringan dapat mendokumentasikan dampak dari serangan ini ke *network bandwidth* dan *availability*, seperti halnya menyediakan informasi pada target yang nyata. *Network traffic data* mungkin juga dapat berguna dalam menginvestigasi aktifitas mencurigakan yang diidentifikasi oleh sumber lainnya. Sebagai contohnya, hal ini mungkin menunjukkan adanya pola komunikasi tertentu yang telah terjadi dalam hari-hari atau minggu-minggu yang terdahulu.
8. Catatan ISP. Informasi dari suatu ISP adalah hal yang terutama pada nilai dalam menyusuri suatu serangan kembali ke sumbernya, yang terutama sekali ketika serangan digunakan untuk alamat *spoofed* IP. Bagian 6.4.4 mendiskusikan hal ini secara mendalam.

#### 6.4.2.2 Tool untuk pengujian

Karena analisis data jaringan dapat dilakukan untuk banyak tujuan dengan lusinan jenis sumber data, analis mungkin dapat menggunakan beberapa tool yang berbeda pada basis regular, masing – masing sesuai untuk situasi tertentu. Analis harus sadar tentang pendekatan yang mungkin untuk menguji data *network traffic* dan harus memilih tool terbaik untuk masing – masing kasus, dibanding menggunakan alat yang sama pada setiap situasi. Analis juga harus sadar tentang kekurangan *tool*; sebagai contoh, *protocol analyzer* tertentu mungkin tidak mampu untuk mengubah protokol tertentu atau menangani data protokol yang tak terduga (misalnya nilai field data yang ilegal). Hal ini akan dapat membantu untuk memiliki persediaan *tool* alternatif yang mungkin tidak memiliki kekurangan yang sama. Analis harus

meninjau ulang hasil yang tidak biasa dan tak terduga yang diproduksi oleh *tool* untuk megkonfirmasikan bahwa mereka valid.

*Tool* sering membantu dalam menyaring data. Sebagai contohnya, seorang analis mungkin perlu untuk mencari data tanpa informasi yang konkrit yang bisa membatasi pencarian tersebut. Hal ini hampir bisa dipastikan untuk terjadi ketika analis bertanggung jawab untuk melakukan tinjauan ulang secara periodik atau berkelanjutan dari kejadian keamanan data log dan tanda (sinyal). Jika volume dari isi *log* dan *alert* sedikit, maka meninjau ulang data secara relatif mudah- tetapi dalam banyak kasus, mungkin ada beribu – ribu daftar kejadian perharinya. Ketika meninjau ulang data manual tidaklah mungkin atau praktis, analis harus menggunakan solusi yang diotomatiskan untuk menyaring kejadian dan memberikan analis kejadian yang hanya mungkin menarik.

Satu teknik efektif adalah mengimpor *log* kedalam database dan menjalankan *query* terhadapnya, jenis penghapusan manapun dari aktifitas yag nampaknya sangat baik dan meninjau ulang sisanya atau memfokuskan pada jenis aktifitas yang nampaknya akan bersifat buruk. Sebagai contoh, jika kecurigaan awalnya adalah server yang telah disetujui melalui aktifitas HTTP, kemudian penyaringan *log* mungkin dimulai dengan penghapusan apapun yang menerima aktifitas HTTP dari pertimbangan. Seorang analis yang sangat terbiasa dengan sumber data tertentu dapat secara umum melakukan pencarian buta sumber data tersebut secara relatif cepat, tetapi pencarian tanpa arah ini dapat memakan banyak waktu pada sumber data yang tidak familiar, karena ada sedikit atau tidak basis untuk menghapus jenis - jenis tertentu aktifitas dari perhatian.

Pilihan lain digunakan untuk *tool* visualisasi, yang mana memberikan data kejadian keamanan dalam format grafik. Hal ini kebanyakan sering digunakan untuk menghadirkan alur *network traffic* secara virtual. Yang mana akan sangat membantu dalam penyelesaian masalah operasional dan mengidentifikasi penyalahgunaan. Sebagai contohnya, *attacker* mungkin menggunakan protokol yang menggunakan *channel* yang tersembunyi dalam cara yang tidak diharapkan untuk mengkomunikasikan informasi secara diam – diam (misalnya pengaturan nilai tertentu dalam protokol jaringan utama atau aplikasi yang menguntungkan). Penggunaan dari *channel* tersembunyi secara umum susah untuk dideteksi, tetapi satu metode yang berguna adalah mengidentifikasi penyimpangan dalam alur *network traffic* yang diharapkan.

*Tool* visualisasi sering tercakup dalam perangkat lunak NFAT, yang diuraikan dalam bagian 6.2.6. Banyak *tool* visualisasi yang dapat melakukan rekontruksi jalur *network traffic* dengan menguji *timestamp field* data sekuensial, *tool* tersebut dapat menentukan urutan dari kejadian dan tampilan secara grafik bagaimana paket melintasi jaringan organisasi. Banyak *tool* visualisasi dapat juga digunakan untuk menampilkan jenis lainnya dari data kejadian keamanan. Sebagai contohnya, seorang analis harus mengimpor catatan pendeteksian gangguan ke tool visualisasi, yang kemudian akan menampilkan data menurut beberapa karakteristik yang berbeda, seperti alamat IP sumber dan tujuan atau *port*. Seorang analis dapat kemudian menyembunyikan tampilan tentang aktifitas baik yang dikenal sehingga hanya aktifitas yang tak dikenal yang ditunjukkan.

Walaupun *tool* visualisasi dapat menjadi sangat efektif untuk meneliti jeni-jenis tertentu dari data, analis secara khusus mengalami suatu kurva pembelajaran secara mendalam dengan *tool* seperti itu. Mengimpor data kedalam *tool* dan menampilkannya adalah hal yang pada umumnya relatif secara langsung, tetapi mempelajari bagaimana menggunakan *tool* secara efisien untuk mengurangi data set yang besar ke sedikit kejadian yang menarik dapat mengambil usaha yang sungguh – sungguh. Rekontruksi *network traffic* mungkin juga dapat dilakukan oleh *protocol analyzer*; meskipun mereka pada umumnya kekurangan kemampuan visualisasi, mereka dapat mengubah paket individual menjadi data *stream* dan menyediakan konteks sekuensial untuk aktifitas.

### 6.4.3 Menggambar kesimpulan

Salah satu yang paling menantang aspek analisa data jaringan adalah tersedianya data yang secara khusus tidak menyeluruh. Dalam banyak kasus, jika bukan kebanyakan, banyak dari data lalu intas jaringan tidak direkam dan sebagi konsekuensinya adalah menghilang. Biasanya analis harus memikirkan tentang proses analisis dalam hubungannya pada pendekatan secara metode yang membangun kesimpulan berdasar pada data yang ada dan asumsi mengenai suatu hilangnya data (yang mana harus berdasrkan pada pengetahuan teknis adn keahlian). Walaupun analis harus bekerja keras untuk menempatkan dan menguji semua data yang tersedia mengenai suatu kejadian, hal ini tidak praktis dalam beberapa hal, terutama sekali ketika ada banyak sumber data yang berlebihan. Selama proses pengujian, analis perlu secepatnya menempatkan, memvalidasi dan meneliti cukup data untuk dapat merekonstruksi suatu kejadian, memahami artinya dan menentukan dampaknya. Dalam banyak kasus, data tambahan tersedia dari sumber jenis lainnya, seperti file data atau sistem

operasi *host*. Bagian menyediakan contoh bagaimana data dari *network traffic* dan sumber lainnya dapat dihubungkan melalui analisis untuk mendapatkan pandangan secara menyeluruh dan akurat tentang apa yang terjadi

Biasanya , analis harus fokus pada mengidentifikasi karakteristik yang penting pada aktifitas dan menilai dampak negatif yang disebabkannya atau mungkin alasan organisasi. Tindakan lainnya, seperti menentukan identitas dari seorang *attacker* eksternal, yang secara khusus *time-intensive* dan sukar untuk menyelesaikan dan tidak membantu dalam mengkoreksi persoalan operasional atau kelemahan keamanan. Menentukan tujuan dari seorang *attacker* juga sangat sulit; sebagai contohnya usaha koneksi yang tidak biasanya dapat disebabkan oleh *attacker*, kode kejahatan, perangkat lunak *misconfigured* atau kesalahan tombol, antar penyebab yang lain. Meskipun tujuan pemahaman penting dalam banyak kasus, dampak negatif dari kejadian harus menjadi perhatian utama. Penetapan identitas dari *attacker* mungkin penting untuk organisasi, terutama ketika aktifitas kriminal telah terjadi, tetapi dalam kasus lainnya hal ini haruslah dipertimbangkan terhadap tujuan penting lainnya untuk meletakkannya dalam gambaran.

Organisasi harus dapat tertarik tidak hanya dalam meneliti kejadian nyata, tetapi juga dalam mengerti penyebab dari tanda bahaya yang palsu . Untuk contohnya, analis sering ditempatkan untuk mengidentifikasi penyebab utama dari positif kesalahan IDS. Ketika sudah pantas, analis harus merekomendasi perubahan ke sumber data kejadian keamanan yang memperbaiki ketelitian pendeteksian.

### 6.4.4 Identifikasi attacker

Ketika meneliti kebanyakan serangan, mengidentifikasi *attacker* bukan perhatian yang segera utama – memastikan bahwa serangan dihentikan dan fokus memulihkan sistem dan data. Jika suatu serangan berkelanjutan, seperti penolakan secara luas dari layanan serangan, organisasi mungkin ingin untuk mengidentifikasi alamat IP yang digunakan oleh *attacker* maka serangan tersebut akan dapat dihentikan. Sayang sekali, hal ini seringkali tidak sesederhana seperti katanya. Materi berikut enjelaskan persoalan potensi melibatkan alam IP yang kelihatannya digunakan untuk melakukan suatu serangan :

1. Alamat IP Gurauan. Banyak serangan yang menggunakan alamat IP gurauan (*spoofed*). Jauh lebih sulit melakukan *spoofing* dengan sukses untuk serangan yang memerlukan koneksi untuk dapat dibangun, maka hal ini pada umumnya digunakan dalam kasus dimana koneksi tidak diperlukan. Ketika paket telah di-*spoof*, biasanya *attacker* tidak

tertarik dalam melihat respon. Hal ini tidak selalu benar—*attacker* dapat melakukan *spoof* pada suatu alamat dari subnet yang mereka monitor, maka ketika tanggapan pergi ke sistem tersebut, mereka dapat men-*sniff* hal itu dari jaringan. Kadang – kadang *spoofing* terjadi oleh suatu kejadian, seperti seorang *attacke*r salah mengkonfigurasi tool dan secara tidak sengaja menggunakan alamat internal NAT. Kadang – kadang suatu gurauan *attacker* yaitu alamat tertentu pada tujuannya, untuk contoh, alamat yang diguraukan mungkin target yang diharapkan diserang dan organisasi melihat kegiatan sederhana dari perantara.

2. Banyak sumber alamat IP. Beberapa serangan nampak untuk menggunakan ratusan atau ribuan dari sumber yang berbeda alamat IP. Kadang – kadang hal ini akurat—sebagai contoh, didtribusikannya penolakan dari serangan *service* secara khusus bersandar pada sejumlah besar yang disepakati mesin yang melakukan serangan terkoordinir. Kadang – kadang hal ini "*bogus*"-serangan yang mungkin tidak memerlukan penggunaan sumber asli dari alamat IP, maka *attacker* menghasilkan banyak IP yang "gadungan" untuk menambahkan kebingungan. Kadang – kadang *attacker* akan menggunakan satu alamat IP asli dan banyak yang palsu; pada kasus tersebut, hal itu memungkinkan untuk mengidentifikasi alamat IP yang sebenarnya dengan mencari aktifitas jaringan lainnya yang telah terjadi sebelum atau sesudah serangan yang menggunakan alamat IP apapun yang sama. Menemukan sesuatu yang sesuai tidak bisa mengkonfirmasikan bahwa alamat tersebut alamat dari *attacker; attacker* dapat tidak hati – hati atau dengan sengaja melakukan gurauan pada alamat IP yang sah yang kebetulan sedang berinteraksi dengan organisasi itu.
3. Validitas alamat IP. Karena alamat IP sering ditempatkan secara dinamis, sistem yang sekarang ini pada alamat IP tertentu tidak mungkin sistem yang sama yang ada disana ketika serangan terjadi, juga, banyak alamat IP yang tidak termasuk kepada pemakai akhir dari suatu sistem, tetapi sebagai gantinya untuk komponen infrastruktur jaringan yang menggantikan alamat IP mereka untuk alamat sumber yang aktual, seperti *firewall* yang menyelenggarakan NAT. Beberapa *attacker* menggunakan *anonymizers*, yang mana antar server yang melaksanakan aktifitas atas nama user untuk memelihara keleluasaan pribadi user.

Berikut ini menguraikan beberapa cara yang mungkin mengusahakan untuk mengesahkan identitas dari suatu *host* yang mencurigakan:

1. Menghubungi pemilik alamat IP. Pendaftaran internet regional, seperti *American Registry for Internet Numbers* (ARIN), menyediakan mekanisme *query* WHOIS pada Web site mereka untuk mengidentifikasi organisasi atau orang yang memilikinya- hal ini bertanggung jawab untuk – alamat IP tertentu. Informasi ini mungkin dapat berguna dalam meneliti beberapa serangan, seperti mengingat tiga alamat IP berbeda yang membangkitkan aktifitas mencurigakan yang semua dicatat kepada pemilik yang sama. Bagaimanapun, dalam kebanyakan kasus, analis tidak harus menghubungi pemiliknya secara langsung; sebagai gantinya, analis harus menyediakan informasi pada pemilik untuk penasehat dan manajemen hukum untuk analis suatu organisasi, yang dapat memulai kontak dengan organisasi atau memberikan persetujuan analis untuk melakukannya jika dibutuhkan. Terutama hal ini seharusnya memperhatikan penyertaan informasi yang dipakai bersama dengan organisasi eksternal; juga, pemilik alamat IP dapat menjadi orang yang melakukan serangan terhadap organisasi tersebut.
2. Mengirimkan *network traffic* ke aamat IP. Organisasi mestinya tidak mengirim jalur *network traffic* untuk menyerang alamat IP sebenarnya untuk memvalidasi identitasnya. Tanggapan apapun yang dihasilkan tidak dapat meyakinkan penegasan identitas dari menyerang host. Jika alamat IP tersebut untuk sistem *attacker, attacker* mungkin melihat jalur lalu lintas dan bereaksi dengan menghancurkan bukti atau menyerang host yang mengirim jalur lalu lintas. Jika alamat IP-nya sudah di "*spoofed*", mengirimkan *network traffic* yang tak diminta ke sistem dapat diartikan sebagai penggunaan yang tidak sah atau suatu serangan. Biar bagaimanapun individu tidak perlu mencoba untuk memperoleh akses ke sistem lainnya tanpa izin.
3. Mencari bantuan ISP. Seperti yang disebutkan didalam bagian 6.3.1, biasanya ISP membutuhkan suatu *court order* sebelum menyediakan informasi apapun ke organisasi pada aktifitas jaringan yang mencurigakan. Maka, bantuan ISP biasanya hanya suatu pilihan selama serangan *network-based* yang paling serius, terutama sekali yang melibatkan gurauan alamat IP. Apakah alamat IP telah di-*spoofe* atau tidak, ISP memiliki kemampuan untuk melacak kembalinya serangan yang berkelanjutan ke sumbernya .
4. Riset dari sejarah alamat IP. Analis dapat mancari aktifitas yang mencurigakan sebelumnya yang dihubungkan dengan alamat IP atau blok alamat IP yang sama. Suatu

organisasi memiliki arsip *network traffic data* dan basis data penelusuran suatu kejadian yang dapat menunjukkan aktifitas sebelumnya. Sumber eksternal yang mungkin meliputi mesin pencari internet dan basis data kejadian online yang mengijinkan pencarian dengan alamat IP.

5. Mencari untuk petunjuk dalam isi aplikasi. paket data aplikasi berhubungan dengan suatu serangan yang mungkin berisi petunjuk untuk identitas *attacker*. Disamping alamat IP, informasi penting lainnya dapat meliputi alamat email atau nama julukan pada Internet relay chat (IRC).

Dalam banyak kasus, organisasi tidak harus secara positif mengidentifikasi alamat IP yang digunakan untuk suatu serangan.

## 6.5 Rekomendasi

Kunci dari rekomendasi dipersembahkan dalam bagian ini untuk menggunakan *network traffic* sebagai berikut :

1. Organisasi perlu memiliki kebijakan mengenai privasi dan informasi yang sensitif. Penggunaan kemampuan *tool* dan teknik untuk analisis data secara tidak hati – hati dapat menyingkapkan informasi sensitif untuk analis dan hal lain yang dilibatkan dalam aktivitas amenganalisis data. Juga penyimpanan jangka panjang dari informasi sensitif yang diambil secara tidak hati – hati oleh *tool* untuk analisa data yang mungkin melanggar kebijakan penyimpanan data. Kebijakan perlu juga alamat yang memonitor jaringan, seperti halnya kebutuhan lambang peringatan pada sistem yang menunjuk aktifitas yang mungkin dimonitor.
2. Organisasi perlu menyediakan penyimpanan yang cukup untuk aktifitas jaringan yang terkait dengan *log*. Organisasi perlu menilai penggunaan puncak dan khusus *log*, menentukan berapa banyak jam atau hari dari data berharga yang harus dipertahankan dan memastikan bahwa aplikasi dan sistem memiliki penyimpanan yang cukup tersedai. *Log* dihubungkan dengan kemampuan peristiwa keamanan komputer yang perlu untuk dijaga untuk suatu periode waktu yang lebih panjang.
3. Organisasi perlu mengatur sumber data untuk meningkatkan kumpulan informasi. Dari waktu ke waktu, pengalaman operasional harus digunakan untuk meningkatkan kemampuan analisa data. Organisasi perlu secara periodik meninjau ulang dan melakukan

penyesuaian dalam menentukan konfigurasi sumber data untuk mengoptimalkan pengambilan informasi yang relevan.

4. Analis perlu mempunyai pengetahuan teknis yang kuat. Sebab *tool* yang sekarang sudah memiliki kemampuan analisa yang agak terbatas, analis perlu untuk dilatih dengan baik;terlatih, berpengalaman, dan banyak mengetahui prinsip *networking*, protokol aplikasi dan jaringan umum, produk keamanan aplikasi dan jaringan, dan ancaman *network-based* dan metode serangan.
5. Analis perlu mempertimbangkan nilai dan ketepatan dari tiap sumber data. Analis perlu mempunyai kepercayaan yang lebih di dalam sumber data asli dibanding sumber data yang menerima data dinormalisir dari sumber lainnya . Analis perlu memvalidasi data manapun yang tak diduga atau tidak biasa berdasarkan pada penelitian atau menginterpretasikan data, seperti IDS dan tanda SEM.
6. Analis biasanya perlu memusatkan pada karakteristik dan dampak dari peristiwa. Menentukan identitas dari suatu *attacker* dan tindakan serupa lain secara khusus *time-intensive* dan sukar untuk memenuhi, dan tidak menopang organisasi dalam mengoreksi kelemahan keamanan atau isu operasional. Menetapkan tujuan dan identitas dari suatu attacker mungkin merupakan hal yang penting, tetapi tujuan lain haruslah dipertimbangkan.

# Penggunaan data dari aplikasi

# 7

Ketenaran dari komputer sebagian besar seharusnya untuk kedalaman dan luasnya aplikasi yang dapat mereka sediakan untuk *user*. Dengan sendirinya, suatu sistem operasi sedikit digunakan; aplikasi yang berjalan pada sistem operasi, seperti email, *Web browser* dan *word processors*, membuat komputer menjadi berharga untuk para *user*. Hal yang sama adalah benar untuk jaringan- mereka yang terutama digunakan untuk mengirimakan aplikasi terkait data antara sistem. File menyediakan mekanisme penyimpanan untuk data aplikasi, menentukan konfigurasi dan *log*. Dari gambaran analisis data, aplikasi membawa bersama file, sistem operasi dan jaringan. Bagian ini menguraikan arsitektur aplikasi—suatu komponen yang secara khusus menyusun aplikasi—dan menyediakan pengertian ke dalam jenis dari aplikasi kebanyakan yang sering fokus terhadap analisa data.

## 7.1 Komponen Aplikasi

Semua aplikasi berisi kode dalam bentuk file *executable* (dan file yang dihubungkan, seperti *code libraries* yang digunakan bersama) atau *script*. Disamping kode, banyak aplikasi yang juga memiliki satu atau lebih komponen tambahan berikutnya : menentukan konfigurasi, otentikasi, *log*, data dan file yang mendukung. Bagian 7.1.1 sampai 7.1.5 menguraikan komponen ini dengan detil dan bagian 7.1.6 mendiskusikan jenis utama dari arsitektur aplikasi, yang mana berhubungan dengan distribusi komponen utama yang diharapkan.

### 7.1.1 Konfigurasi Setting

Kebanyakan aplikasi yang mengijinkan *user* atau *administrator* untuk mengkostumisasi aspek tertentu dari perilaku aplikasi dengan merubah *configuration settings*. Dari gambaran analisis data, banyak peraturan yang biasanya sepele (misalnya menspesifikasi warna *background*), tetapi lainnya mungkin lebih penting, seperti host dan direktori dimana file data dan *log* disimpan, atau *default username*. *Configuration settings* mungkin bersifat sementara—mengatur secara dinamis selama sesi aplikasi tertentu—atau permanen. Banyak aplikasi yang memiliki beberapa pengaturan yang berlaku untuk semua users dan juga

mendukung beberapa pengaturan *user-spesific*. *Configuration settings* mungkin disimpan dalam beberapa cara, mencakup berikut:

1. Konfigurasi file

   Aplikasi dapat menyimpan pengaturan dalam file teks atau suatu file dengan kepemilikan format biner. Beberapa aplikasi memerlukan file konfigurasi untuk dapat menjadi aplikasi pada host yang sama, ketika aplikasi lainnya mengijinkan file konfigurasi untuk dapat ditempatkan pada host lainnya. Sebagai contohnya, suatu aplikasi mungkin diinstal pada suatu workstation, tetapi file konfigurasi untuk user tertentu dapat disimpan pada direktori utama user pada suatu file server.

2. Runtime Option

   beberapa aplikasi mengijinkan *configuration settings* tertentu untuk dapat dispesifikasikan pada saat *runtime* melalui penggunaan pilihan *command-line.* Sebagai contohnya, Unix *e-mail client mutt* memiliki pilihan untuk menspesifikasi lokasi dari *mailbox* untuk dibuka dan lokasi dari file konfigurasi. Mengidentifikasi pilihan mana yang digunakan untuk sesi aktif adalah aplikasi spesifik dan OS; metode yang mungkin meliputi peninjauan ulang daftar proses OS yang aktif, menguji file history suatu OS dan meninjau ulang suatu log aplikasi. pilihan *runtime* dapat juga dispesifikasikan dalam *icon, startup file, batch files* dan cara lainnya.

3. Yang ditambahkan ke sumber program

   beberapa aplikasi membuat kode sumber tersedia (misalnya aplikasi *open source, script*) yang benar – benar menempatkan *user* atau *admisnistrator-specified configuration settings* yang secara langsung kedalam sumber kode. Jika aplikasi kemudian di-*compile*(misalnya kode *human-readable* dikonversikan ke biner, format *machine readable*), i yang mungkin benar – benar dimasukkan tak lebih dari file *executable*, berpotensi membuat pengaturan yang jauh lebih sulit untuk diakses dibanding jika mereka dispesifikasi dalam file konfigurasi atau pilihan *runtime*.

### 7.1.2 Pengesahan (Otentikasi)

Beberapa aplikasi memverifikasi identitas dari tiap pemakai yang mencoba untuk menjalankan aplikasi tersebut. Walaupun ini pada umumnya dilaksanakan untuk mencegah akses tidak sah terhadap aplikasi, hal ini mungkin juga dilaksanakan ketika akses tidak diperhatikan sedemikian sehingga aplikasi dapat di-customize didasarkan pada identitas para user. Metode pengesahan yang umum meliputi yang berikut:

1. Pengesahan Eksternal. Aplikasi mungkin menggunakan layanan otentikasi eksternal, seperti direktori suatu server. Meskipun aplikasi berisi banyak catatan yang berhubungan dengan otentikasi, layanan otentikasi eksternal mungkin berisi informasi otentikasi yang lebih detil.
2. Pengesahan Kepemilikan. Aplikasi mungkin memiliki mekanisme pengesahan sendiri, seperti kata sandi dan *user accounts* yang menjadi bagian dari aplikasi tersebut, bukan sistem operasinya.
3. *Pass-Through Authentication*. *Pass-through authentication* mengacu pada menghantar surat kepercayaan sistem operasi (secara khusus, *username* dan *password*) yang tidak dienkripsi dari sistem operasi ke aplikasi.
4. *Host/user Environmnet*. Di dalam suatu lingkungan yang dikendalikan (misalnya *workstation* dan *server* yang diatur didalam suatu organisasi), beberapa aplikasi mungkin mampu bersandar pada otentikasi sebelumnya yang dilakukan oleh OS. Sebagai contohnya, jika semua *host* menggunakan suatu aplikasi yang merupakan bagian dari *domain* Windows yang sama, dan masing – masing *user* telah siap diotentikasi oleh *domain*, maka aplikasi dapat mengekstrak identitas *OS-authenticated* oleh domain dari tiap lingkungan *workstation*. Suatu aplikasi dapat membatasi akses ke aplikasi dengan menelusuri user mana yang diijinkan mengakses dan membandingkan identitas *OS-authenticated* ke daftar *user* yang diijinkan. Teknik ini hanya efektif jika *user* tidak dapat mengubah identitas *user* dalam lingkungan *workstation*.

Implementasi pengesahan tertukar – tukar secara luas antara aplikasi dan lingkungan, dan hal tersebut diluar lingkup dari dokumen ini untuk didiskusikan dengan detil. Analis harus sadar bahwa ada banyak cara yang berbeda dalam user mana yang dapat diotentikasikan. Maka, sumber dari catatan otentikasi user mungkin sangat dapat tertukar diantara aplikasi dan implementasi aplikasi. analis harus juga mengetahui bahwa beberapa aplikasi menggunakan kontrol akses (secara khusus dijalankan oleh sistem operasi) dalam membatasi akses untuk informasi dan fungsi aplikasi jenis tertentu. Hal ini dapat membantu dalam menentukan aplikasi user tertentu apa yang bisa dipergunakan. Beberapa aplikasi merekam informasi yang berhubungan dengan kontrol akses, seperti usaha yang digagalkan untuk melakukan tindakan sensitif atau akses data yang dibatasi

### 7.1.3 Logs

Meskipun beberapa aplikasi (terutama salah satunya yang sederhana) tidak merekam banyak informasi ke *log*, kebanyakan aplikasi melakukan beberapa jenis *logging*.Suatu aplikasi mungkin merekam *log entries* ke log OS yang spesifik (misalnya *syslog* pada sistem Unix, *event logs* pada Windows sistem), file teks, database atau kepemilikan format file. Beberapa aplikasi merekam kejadian dengan jenis berbeda ke log yang berbeda. Jenis umum dari *log entries* adalah sebagai berikut :

1. Peristiwa, *event log entries* secara khusus mendaftarkan tindakan yang dilakukan, tanggal dan waktu tiap tindakan yang terjadi dan hasil dari tiap – tiap tindakan. Contoh dari tindakan yang mungkin direkam adalah menetapkan koneksi ke sistem lainnya dan mengeluarkan perintah *administrator-level*. *Event log entries* mungkin juga meliputi mendukung informasi, seperti *username* apa yang digunakan untuk melakukan masing-masing tindakan dan kode status apa yang dikembalikan ( yang mana menyediakan lebih banyak informasi pada hasil dibanding suatu status sederhana yang sukses atau gagal).
2. Audit. *Audit log entries* juga dikenal sebagai *security log entries*, berisi informasi yang menyinggung aktifitas yang diaudit, seperti sukses atau gagalnya mencoba untuk *logon*, perubahan kebijakan keamanan, akses file dan proses eksekusi. Aplikasi mungkin menggunakan kemampuan audit membangun sistem operasi atau menyediakan kemampuan mengaudit mereka sendiri.
3. Kesalahan(*Error*). Beberapa aplikasi membuat catatan kesalahan, yang mana merekam informasi mengenai kesalahan aplikasi, secara khusus dengan *timestamps*. *Error log* dapat membantu dalam penyelesaian kedua masalah yaitu operasional dan serangan. *Error messages* menentukan kapan kejadian yang penting terjadi dan mengidentifikasi beberapa karakteristik suatu kejadian.
4. Instalasi. Aplikasi mungkin membuat file *log* instalasi terpisah yang merekam informasi yang berhubungan kepada instalasi awal dan aplikasi yang diperbaharui berikutnya. Informasi direkam dalam *log* instalasi yang bervariasi secara luas, tetapi mungkin untuk mencakup status pada berbagai fase instalasi dan mungkin juga untuk menunjuk sumber dari file instalasi, lokasi dimana komponen aplikasi ditempatkan dan pilihan yang melibatkan konfigurasi suatu aplikasi.
5. *Debugging*. Beberapa aplikasi dapat dijalankan dalam mode *debugging*, yang mana berarti bahwa *log* mereka jauh lebih banyak informasi dibanding mengenai operasi umum dari

aplikasi. Catatan *debugging* sering sangat *cryptic* dan mungkin hanya memiliki arti ke pembuat perangkat lunak, yang dapat menerjemahkan kode kesalahan dan segi lainnya menyangkut catatan tersebut. Jika suatu aplikasi menawarkan kemampuan *debugging*, secara khusus hal itu hanya memungkinkan jika *administrator* atau *developer* perlu untuk memecahkan suatu masalah operasional yang spesifik.

### 7.1.4 Data

Hampir tiap – tiap aplikasi secara spesifik didesain untuk menangani data dalam satu atau lebih cara, seperti *creating, displaying, transmitting, receiving, modifying, deleting, protecting* dan *storing* data. Sebagai contoh, suatu *e-mail client* mengijinkan *user* untuk membuat pesan e-mail dan mengirimkannya ke seseorang, dan untuk menerima, melihat dan menghapus pesan email dari orang lain. Data aplikasi sering berada untuk sementara dalam memori dan bersifat sementara atau permanen dalam file. Format dari suatu file yang berisi data aplikasi yang mungkin umum (misalnya file teks, grafik bitmap) atau kepemilikan. Data mungkin juga disimpan dalam basis data, yang mana merupakan kumpulan file dan spesifikasi data yang sangat terstruktur . beberapa aplikasi membuat file sementara selama suatu sesi, yang mana mungkin berisi data aplikasi. jika suatu aplikasi gagal untuk dimatikan , hal itu mungkin meninggalkan file sementara pada suatu media. Kebanyakan sistem operasi memiliki direktori yang didesain untuk file sementara; bagaimanapun, beberapa aplikasi memiliki direktori sementara sendiri dan aplikasi lainnya menempatkan file sementara dalam direktori yang sama dimana data disimpan. Aplikasi mungkin juga berisi template file data dan contoh file data (misalnya basis data, dokumen)

### 7.1.5 Pendukung file

Aplikasi sering mencakup satu atau lebih jenis pendukung file, seperti dokumentasi dan grafik. Pendukung file cenderung untuk menjadi statis, tetapi hal itu bukan berarti mereka tidak penting untuk analisa data. Jenis pendukung file meliputi yang berikut ini :

1. Aplikasi File sering meliputi satu atau lebih jenis pendukung file, seperti grafik dan dokumentasi.
2. Dokumentasi. Hal ini meliputi *administrator* dan *user manual, help files*, dan perijinan informasi. Dokumentasi dapat membantu untuk menganalisa dalam banyak cara, seperti menjelaskan apa yang dikerjakan suatu aplikasi, bagaiman aplikasi berkerja dan komponen apa yang aplikasi miliki. Dokumentasi juga secara khusus berisi informasi

kontak untuk *vendor* suatu aplikasi; suatu *vendor* mungkin dapat menjawab pertanyaan dan menyediakan bantuan lainnya dalam pemahaman tentang aplikasi tersebut.

3. *Links.* Juga dikenal sebagai *shortcuts, links* sederhananya adalah suatu penunjuk ke hal lain, seperti suatu *executable. Links* adalah yang paling sering digunakan dalam sistem Windows; sebagai contohnya, daftar materi pada menu Start sungguh dapat menghubungkan ke program. Dengan menguji properti dari *link*, seorang analis dapat menentukan program apa yang dijalankan *link,* dimana programnya dan pilihan apa yang diatur (bila ada).
4. Grafik. hal ini meliputi grafik "*standalone*" yang digunakan oleh aplikasi, seperti halnya grafik untuk *icon*. Meskipun aplikasi grafik secara khusus sedikit menarik untuk analis, grafik *icon* mungkin dapat menarik dalam mencoba untuk mengidentifikasi *executable* mana yang sedang dijalankan.

### 7.1.6 Arsitektur aplikasi

Setiap aplikasi memiliki arsitektur, yang mana mengacu pada logis tersendiri dari komponen dan penggunaan mekanisme komunikasi diantara komponen. Kebanyakan aplikasi didesain untuk mengikuti salah satu dari tiga kategori arsitektur aplikasi utama, sebagai berikut :

1. Lokal. Aplikasi lokal dimaksudkan untuk dapat diisi sebagian besar dalam sistem tunggal. Kodenya, *configuration settings, logs* dan pendukung file yang ditempatkan pada sistem *user*. Aplikasi lokal tidak mungkin melakukan otentikasi. Data aplikasi mungkin diisi pada sistem *user* atau sistem lainnya (misalnya file server ) dan pada umumnya tidak dapat dimodifikasi secara serempak oleh banyak *user*. Contoh aplikasi lokal adalah *text editors, graphics editors,* and *office productivity suites* (misalnya *word processor, spreadsheet*).
2. *Client/Server.* Suatu *client/server* aplikasi dirancang untuk dapat dipisah diantara berbagai sistem. Suatu aplikasi *two-tiered client/server* menyimpan kodenya, *configuration setting* dan pendukung file pada masing – masing *workstation* dan datanya pada satu atau lebih pusat server yang diakses oleh semua user. *Log* hampir bisa dipastikan disimpan pada workstation saja. Aplikasi *two-tiered client/server* memisahkan *user interface* dari sisa aplikasi dan juga memisahkan data dari komponen lainnya. Model *three-tier* yang klasik menempatkan kode *user interface* pada *client workstation* (bersama dengan beberapa file pendukung), sisa dari kode aplikasi pada suatu server aplikasi dan server basis data.

Masing – masing strata hanya berinteraksi dengan strata yang bersebelahan, maka dalam model tiga dan empat strata, *client* tidak secara langsung berinteraksi dengan server basis data. Contoh khusus aplikasi *client/server* adalah sistem pencatatan kesehatan, aplikasi *e-commerce* dan sistem yang menginventarisir.

3. *Peer-To-Peer*. Aplikasi *peer-to-peer* didesain sedemikian sehingga individu *client host* berkomunikasi secara langsung dan membagi data satu sama lainnya. Secara khusus, komunikasi pertama client dengan server yang dipusatkan itu menyediakan informasi pada klien lainnya; informasi ini kemudian digunakan untuk menetapkan koneksi langsung yang tidak membutuhkan dipusatkannya server sampai berhasil. Contohnya aplikasi *peer-to-peer* yang merupakan file bersama tertentu, *cht* dan program *instant messaging*. Sebagian dari program ini dikenal sebagai *peer-to-peer* tetapi secara aktual merupakan *client/server*, karena komunikasi *client* dengan *server* yang dipusatkan, sebagai ganti berkomunikasi secara langsung dengan satu sama lainnya.

Kebanyakan aplikasi sangat fleksibel dalam kaitannya denagn arsitektur. Sebagai contohnya, banyak aplikasi *client/server* dapat memiliki banyak strata pada sistem tunggal. Terutama selama demo atau percobaan aplikasi, semua strata mungkin diinstal pada satu sistem. Pada sisi lain, beberapa aplikasi lokal dapat dipisah antar sistem, dengan beberapa komponen pada sistem lokal dan beberapa pada sistem *remote*. Aplikasi sering membuat hal tersebut mudah untuk menspesifikasi dimana komponen berbeda harus diinstal dan dimana file data dan konfigurasi harus disimpan. Untuk banyak aplikasi, hal tersebut merupakan perlakuan yang baik dari varietas antar penyebaran.

Aplikasi yang dirancang untuk memisahkan kode mereka antar banyak host yang secara khusus menggunakan protokol aplikasi untuk komunikasi diantara host. Jenis aplikasi yang ada dimanapun seperti e-mail dan pengunaan Web yang dikenal, protokol aplikasi yang distandarisasi untuk memfasilitasi interoperabilitas antar komponen yang berbeda. Sebagai contoh, hampir setiap program *e-mail client* kompatibel dengan hampir setiap program server e-mail karena mereka didasarkan pada protokol aplikasi standar yang sama. Bagaimanapun, suatu program yang berdasar pada standar dapat menambahkan *fitur proprietary* atau melanggar standar dalam beberapa cara, khususnya jika standar tidak dirinci secara mendalam. Jika interoperabilitas dengan aplikasi lainnya yang tidak diperhatikan (atau tidak

dinginkan ) dan bagian yang sama membuat semua komponen aplikasi, protokol bukan standar yang sering digunakan.

Seperti yang diuraikan melalui bagian 7.1, aplikasi mungkin memiliki banyak perbedaan komponen yang beroperasi bersama – sama. Sebagai tambahan, suatu aplikasi mungkin bergantung pada satu atau lebih aplikasi lainnya. sebagai contoh, banyak aplikasi *e-commerce* dijalankan *client* di dalam *Web Browsers*. Banyak aplikasi yang juga bersandar pada layanan OS, seperti pencetakan dan DNS *lookups* (untuk mencari alamat IP dari server aplikasi dan alat lainnya). Aplikasi tertukar – tukar secara luas dalam kompleksitas, dari program untuk fungsi sederhana seperti kalkulator untuk aplikasi *e-commerce* yang besar dimana dapat melibatkan ribuan dari komponen dan jutaan *user*.

## 7.2 Jenis dari aplikasi

Aplikasi ada untuk hampir setiap tujuan dapat dibayangkan. Walaupun teknik analisa data dapat diaplikasikan ke banyak aplikasi, aplikasi tertentu dari aplikasi mungkin dapat menjadi fokus dari analisis termasuk e-mail, penggunaanWeb, pesan interaktif, pemakaian file secara bersama, penggunaan dokumen, aplikasi keamanan dan tool persembunyian data. Hampir setiap komputer memiliki sedikitnya beberapa aplikasi yang diinstal dari kategori ini. Bagian berikut menguraikan masing-masing jenis aplikasi ini secara lebih detil.

### 7.2.1 E-mail

E-mail telah menjadi alat – alat yang utama untuk orang berkomunikasi secara elektronis. Masing – masing pesan e-mail terdiri dari *header* dan *body*. *Body* dari e-mail terdiri dari isi yang aktual dari pesan, seperti sebuah memo atau surat personal. *Header* dari e-mail meliputi berbagai macam bagian dari informasi mengenai e-mail. Dengan kondisi *default*, kebanyakan aplikasi *e-mail client* hanya menampilkan sedikit *header field* untuk masing – masing pesan : alamat e-mail pengirim dan yang dikirim, tanggal dan waktu pesan ketika dikirimkan dan subjek dari pesan. Bagaimanapun, *header* secara khusus mencakup beberapa *field* lainnya, meliputi hal yang dibawah ini :

1. ID pesan.
2. Jenis dari *e-mail client* yang digunakan untuk membuat pesan.
3. Pentingnya pesan yang ditunjukkan oleh pengirim (misalnya rendah, normal, tinggi).
4. Mengirimkan informasi—yang mana server *e-mail* suatu pesan dilewati dalam pemindahan dan ketika masing – masing server tersebut diterima.

5. Jenis isi pesan, yang mana menunjukkan apakah *e-mail* dengan isi yang sederhana terdiri dari suatu badan teks atau juga mempunyai file *attachment*, *embedded graphics*, dan lain lain.

Aplikasi *e-mail client* digunakan untuk menerima, menyimpan, membaca, menyusun, dan mengirimkan e-mails. Kebanyakan *e-mail client* juga menyediakan suatu buku alamat yang dapat menjaga informasi kontak, seperti alamat e-mail, nama, dan nomor telepon. Program yang dienkripsi kadang – kadang digunakan bersama – sama dengan *e-mail client* untuk mengenkripsi suatu e-mail *body* dan atau *attachment*.

Ketika seorang *user* mengirim suatu *e-mail*, hal itu ditransfer dari e-mail *client* ke *server* menggunakan SMTP. Jika pengirim dan orang yang dituju dari *email* menggunakan *e-mail server* yang berbeda, *e-mail* kemudian diarahkan menggunakan SMTP melalui *e-mail server* tambahan sampai hal tersebut menjangkau server dari orang yang dituju. Secara khusus, *recipient* menggunakan *e-mail client* pada sistem yang terpisah untuk mendapatkan kembali *e-mail* menggunakan *Post Office Protocol 3* (POP3) atau Internet *Message Access Protocol* (IMAP); dalam banyak kasus, *e-mail client* mungkin terdapat pada server tujuan (misalnya sistem Unix untuk *multi-user*). Tujuan server sering melakukan pemeriksaan pada *e-mails* sebelum membuat mereka tersedia untuk diperoleh kembali, seperti memblok pesan dengan isi yang tidak sesuai (misalnya spam, virus). Dari akhir sampai akhir lagi, informasi mengenai pesan e-mail tunggal mungkin direkam dalam beberapa tempat—sistem pengirim, masing –masing server *e-mail* yang menangani pesan dan sistem *recipient*, seperti halnya antivirus, *spam* dan isi yang menyaring server.

### 7.2.2 Penggunaan Web

Melalui *Web browsers*, orang mengakses *Web servers* yang berisi hampir semua jenis data yang dapat dibayangkan. Banyak aplikasi juga menawarkan *interface Web-based*, yang mana juga diakses melalui *Web browsers*. Karena mereka dapat digunakan untuk banyak tujuan, *Web browser* adalah satu dari kebanyakan aplikasi umum yang digunakan. Standar dasar untuk komunikasi Web adalah *HyperText Transfer Protocol* (HTTP); bagaimanapun, HTTP mungkin dapat berisi banyak jenis dari data dalam variasi standar dan format kepemilikan. Hal utama HTTP hanya mekanisme untuk memindahkan data diantara *Web browsers* dan *Web servers.*

Secara khusus, sumber informasi yang paling kaya mengenai penggunaan Web adalah *host* yang sedang menjalankan *Web browser*. Informasi mungkin dapat diterima kembali dari *Web browsers* meliputi daftar *Web site* yang favorit, *history* (dengan *timestamp*) dari *Web site* yang didatangi, file data *cached* Web dan *cookie* (meliputi tanggal dibuat mereka dan waktu habisnya). Sumber baik dari informasi penggunaan Web lainnya adalah *Web servers*, yang mana secara khusus menjaga *log* menyangkut permintaan yang terima. Data sering di*log* oleh *web server* untuk masing – masing permintaan meliputi timestamp; alamat IP, versi *Web browser* dan OS dari *host* yang membuat permintaan; jenis dari permintaan (misalnya membaca data, menulis data); dimintanya sumber daya; dan kode status. Tanggapan untuk masing – masing permintaan meliputi tiga digit kode status yang menunjukkan sukses atau gagalnya suatu permintaan. Untuk sukses, kode status menjelaskan aksi apa yang dilakukan; untuk gagal, kode status menjelaskan mengapa permintaan digagalkan.

Di samping *Web browser* dan server, beberapa jenis alat dan perangkat lunak lainnya mungkin juga mencatatkan informasi yang terkait. Sebagai contohnya, *Web proxy server* dan aplikasi yang mewakili *firewalls* mungkin melakukan pencatatan secara detil tentang sktifitas HTTP, dengan level yang serupa dari detil untuk log *Web server*. *Routers, non-proxying firewalls* dan alat jaringan lainnya mencatat aspek dasar dari koneksi jaringan HTTP, seperti alamat dan *port* dari sumber dan tujuan. Organisasi menggunakan *Web content monitoring* dan *filtering service* untuk menemukan kegunaan data dalam *service log*, terutama mengenai penolakan permintaan Web.

### 7.2.3 Komunikasi interaktif

Tidak seperti pesan e-mail, yang mana secara khusus mengambil beberapa menit untuk meninggalkan pengirim ke *recipient*, layanan komunikasi interaktif menyediakan waktu komunikasi (atau mendekati waktu sebenarnya) sebenarnya. Jenis aplikasi yang biasanya digunakan untuk komunikasi interaktif meliputi hal berikut ini :

1. *Group chat*. Aplikasi *group chat* menyediakan tempat pertemuan virtual dimana banyak *user* dapat berbagi pesan dalam satu waktu. Secara khusus aplikasi *group chat* menggunakan arsitektur *client/server* . Protokol *group chat* yang paling populer adalah *Internet Relay Chat* (IRC) yang merupakan protkol standar yang menggunakan komunikasi *text-based* secara relatif sederhana. IRC juga menyediakan mekanisme untuk *user* mengirim dan menerima file.

2. *Instant Messaging Applications*. *Instant Messaging* (IM) *applications* adalah salah satu dari *peer-to-peer*, membiarkan user untuk mengirimkan pesan teks dan file secara langsung untuk satu dengan yang lainnya atau *client /server*, mengantarkan pesan dan file melalui server yang dipusatkan. *Configuration setting* dari aplikasi IM mungkin berisi informasi *user,* daftar dari *user* dimana *user* dapat berkomunikasi didalamnya, informasi file yang dipindahkan dan pesan yang disimpan atau sesi chat. Ada beberapa layanan *Internet-based* IM yang utama, masing –masing penggunaan yang mana adalah protokol komunikasi kepemilikan sendiri. Beberapa perusahaan juga menawarkan perusahaan produk IM yang dijalankan dalam suatu organisai. Seperti produk yang sering diintegrasikan sampai taraf tertentu dengan layanan *e-mail* organisasi dan hanya dapat digunakan oleh *e-mail user* yang diotentikasi.
3. *Video* dan *audio*. Sebagai kapasitas dari jaringan yang dilanjutkan untuk ditingkatkan, pelaksanaan komunikasi *real-time video* dan *audio* ke jaringan komputer lainnya telah menjadi sesuatu yang umum. Teknologi seperti *Voice over IP* (VoIP) mengijinkan orang untuk melakukan percakapan dari telepon di jaringan seperti internet. Banyak implementasi audio yang menyediakan layanan *computer-based* dari akhir sampai kahir, ketika lainnya hanya secara parsial berbasis komputer, dengan server antara yang mengubah komunikasi antara jaringan komputer dan jaringan telepon standar. Banyak teknologi audio , terutama plikasi *peer-to-peer*. Teknologi video dapa digunakan untuk melakukan *teleconference* atau memiliki komunikasi " *video phone*" meliputi H.323 dan *Session Initiation Protocol* (SIP).

### 7.2.4 File yang digunakan bersama – sama

*User* dapat men-*share* file dengan menggunakan banyak program berbeda. Seperti yang diuraikan sebelumnya dalam bagian ini, e-mail, program *group chat* dan semua perangkat lunak IM yang menyediakan kemampuan untuk mengirim dam menerima file tertentu. Bagaimanapun, program ini pada umumnya tidak mengijinkan *recipient* untuk mem-*browse* melalui file dan memilih file untuk dipindahkan.Program penggunaan *file fullfledged* secara bersama dan protokol yang dibutuhkan untuk level kemampuan ini. Program untuk penggunaan file bersama dapat dikelompokkan oleh arsitektur, seperti berikut ini :

1. *Client/Server*. Layanan file *sharing* tradisional menggunakan arsitektur *Client/Server*, dengan server pusat file yang berisi tempat penyimpanan file. *Client* dapat menggunakan *server* dengan mengawali koneksinya, mengotentikasi (jika dibutuhkan), meninjau ulang

daftar dari file yang tersedia (jika diperlukan) kemudian memindahkan file dari atau untuk server. Contohnya layanan *client/server* file sharing yang biasa digunakan adalah *File Transfer Protocol* (FTP), *Network File Sharing* (NFS), *Apple Filing Protocol* (AFP), *Message Block* (SMB). Hal ini merupakan protokol yang distandarisasi bahwa tidak melindungi kerahasiaan dari data dalam pemindahan, meliputi banyak *authentication credentials* yang disediakan seperti *password*, alternatif pengamanan, seperti *Secure* FTP (SFTP) dan *Secure* Copy (scp), mengenkripsikan jaringan komunikasi mereka. Kebanyakan sistem operasi memiliki file *sharing clients* didalamnya (misalnya FTP, SMB), tetapi *user* dapat juga menginstal program *third-party* yang bervariasi yang menyediakan kemampuan yang serupa.

2. *Peer-To-Peer.* Kebanyakan layanan *peer-to-peer file sharing* yang terutama digunakan untuk tukar menukar musik, grafik atau perangkat lunak lewat internet. Tidak seperti *client/server file sharing*, dimana server tunggal menjaga tempat penyimpanan file, *peer-to-peer file sharing* didistribukan, dengan dialokasikannya file pada banyak *host* yang berbeda. Layanan *client/server file sharing* secara khusus memiliki server utama yang memberi informasi *client* pada saat dimana *client* yang lain dialokasikan, tetapi server tidak ikut andil dalam transmisi file atau infoemasi file. Layanan *peer-to-peer file sharing* secara tidak diperlukan otentikasi *user*. Semua file yang dibrowse dan dipindahkan terjadi secara langsung diantara *clients (peers).* Secara khusus *user* dapat memilih dari beberapa program *client* ketika menggunakan layanan tertentu. Meskipun kebanyakan layanan mengijinkan masing – masing *user* untuk mengkontrol file mana yang di*share* pada sistem mereka, layanan dikenal sebagai pekerjaan *peer-to-peer* yang dienkripsi oleh penyimpanan file lainnya pada bagian yang dienkripsi pada masing – masing *hard drive user* dan tidak memberikan kontrol *user* yang berlebihan atau pengetahuan tentang apa yang disimpan dalam area tersebut dari sistem mereka sendiri .Tanpa layanan *peer-to-peer* mengirim file yang diminta melalui banyak *host* , sebagai gantinya mengirimkan mereka secara sederhana dari sumber ke tujuan, dengan tujuan membuatnya lebih sulit untuk diidentifikasi sumber sebenarnya atau tujuan dari file apapun yang diberikan.

### 7.2.5 Penggunaan dokumen

Banyak user yang menggunakan waktu mereka untuk bekerja dengan dokumen, seperti surat, laporan dan tabel. Dokumen mungkin berisi tipe dari data manapun, maka mereka merupakan hal yang sering menarik untuk analis. Kelas dari perangkat lunak

digunakan untuk membuat, mengamati dan mengedit dokumen seperti itu yang dikenal sebagai aplikasi produktivitas kantor. Didalmnya termasuk perangkat lunak *word processor*, *spreadsheet, presentation* dan basis data pribadi. Dokumen seringkali memiliki *user* atau informasi sistem yang bersifat *embedded* didalamnya, seperti nama atau *username* dari orang yang membuat atau terakhir mengedit suatu dokumen atau nomor lisensi dari perangkat lunak atau alamat MAC dari sistem yang digunakan untuk membuat dokumen.

### 7.2.6 Aplikasi keamanan

*Host* sering menjalankan satu atau lebih aplikasi keamanan yang berusaha melindungi host dari penyalahgunaan yang terjadi melalui aplikasi yang biasanya digunakan, seperti *e-mail client dan Web browser*. Contoh dari aplikasi keamanan yang biasanya digunakan meliputi perngakt lunak antivirus, pendeteksi *spyware* dan *removal utilities*, isi yang menyaring (misalnya *anti-spam measures*) dan perangakat lunak pendeteksi gangguan *host-based*. *Log* dari aplikasi keamanan mungkin berisi catatan detil dari aktifitas yang mencurigakan dan mungkin menunjukkan apakah keamanan yang disetujui terjadi atau dicegah. Jika aplikasi keamanan adalah bagian dari penyebaran suatu perusahaan , seperti dikendalikan atau dikontrol perangkat lunak antivirus, *log* mungkin dapat keduanya tersedia pada *host* individu dan pada log aplikasi yang dipusatkan.

## 7.2.7 Data Concealment Tools

Beberapa orang menggunakan *tool* untuk merahasiakan data dari hal lainnya. Hal ini mungkin dilakukan untuk tujuan yang baik, seperti melindungi integritas dan kerahasiaan data terhadap akses oleh bagian yang tidak sah atau untuk tujuan kejahatan seperti merahasiakan bukti aktifitas yang tidak pantas. Contoh *tool* untuk merahasiakan data meliputi utilitas enkripsi file, *tool* steganografi dan *tool* untuk membersihkan sistem yang merupakan tujuan khusus perangkat lunak untuk membuang data yang menyinggung aplikasi tertentu, seperti Web browser, seperti halnya didalam lokasi umum, seperti direktori sementara. Penggunaan kebanyakan tool untuk merahasiakan data tidak mungkin untuk diambil didalam *log*. Analis harus sadar akan kemampuan *tool* ini sedemikian sehingga mereka dapat mengidentifikasi seperti *tool* pada sistem dan mengenal efek dari *tool.*

## 7.3 Mendapatkan data aplikasi

Seperti yang diuraikan dalam bagian 7.1, data terkait dengan aplikasi mungkin dapat ditempatkan dalam sistem file, data OS yang bersifat volatil dan *network traffic*. Bagian 4.2, 5.2 dan 6.3 berisi informasi yang spesifik pada perolehan data dari sumber masing – masing. Daftar berikut merupakan tipe data aplikasi yang masing – masing sumber yang mungkin berisi:

1. Sistem file. Sistem file mungkin berisi banyak tipe file yang berhubungan dengan aplikasi, meliputi file *executable* dan *script*, file konfigurasi, pendukung file (misalnya dokumentasi), *log* dan file data.
2. Data OS yang bersifat volatil. Data OS yang bersifat volatil mungkin berisi informasi pada koneksi jaringan yang digunakan oleh aplikasi, jalannya proses suatu aplikasi pada suatu sistem dan *argumen command line* digunakan untuk masing – masing proses dan file yang dipegang dibuka oleh aplikasi, seperti halnya jenis pendukung informasi lainnya.
3. *Network Traffic*. Kebanyakan *network traffic data* yang relevan melibatkan hubungan *user* untuk aplikasi *remote*. Dan komponen aplikasi pada sistem berbeda berkomunikasi satu dengan yang lainnya. Catatan *network traffic* lainnya mungkin juga menyediakan pendukung informasi, seperti koneksi jaringan untuk mengendalikan pencetakan dari suatu aplikasi dan DNS *looksup* oleh aplikasi *client* atau komponen lainnya untuk memecahkan komponen aplikasi nama *domain* ke alamat IP.

Analis sering meghadapi tantangan utama dalam menentukan data mana yang harus diperoleh. Dalam banyak kasus, analis pertama kali harus memutuskan aplikasi mana yang menarik. Sebagai contohnya, hal ini biasanya memiliki banyak *Web browser* dan *e-mail client* yang diinstal pada sistem tunggal. Jika analis ditugaskan untuk memperoleh data mengenai penggunaan individu dari layanan *e-mail* organisasi, mereka perlu berhati – hati pada semua cara yang mana individu perlu diakses layanan tersebut .*User* komputer perlu mengisi tiga *e-mail client* yang berbeda, ditambah dua *Web browser* yang perlu digunakan untuk mengakses *Web-based e-mail client* yang disediakan oleh organisasi.Untuk *user* komputer, analis dapat memperoleh dengan sederhana semua data darinya dan kemudian menentukan *client* mana yang benar - benar digunakan untuk e-mail selama proses pengujian. Bagaimanapun ada banyak sumber data yang berpotensi disamping *user* komputer dan sumber ini mungkin

ditukar – tukar berdasarkan pada *client* yang digunakan. Untuk contohnya, penggunaan *Web-based client* mungkin telah direkam dalam server Web, *firewall,* IDS, IDS dan *content* yang memonitor *log* perangkat lunak,seperti halnya *Web browser history file* ,*Web browser caches, cookies* dan *personal firewalls*. Dalam beberapa situasi, memperoleh data yang diperlukan mungkin melibatkan identifikasi semua komponen dari aplikasi, memutuskan data dari komponen tersebut. Bagian 8 berisi contoh yang menggambarkan kompleksitas dalam mengidentifikasi komponen dan memprioritaskan memperoleh data untuk aplikasi.

## 7.4 Menguji Data Aplikasi

Sebagian besar data aplikasi yang spesifik yang terdiri dengan memperhatikan bagian – bagian yang spesifik dari data palikasi—sistem file, data OS yang volatil dan *network traffic*—menggunakan teknik dan *tools* yang berturut -turut diuraikan dalam bagian 4.3, 5.3, and 6.4. Pengujian mungkin dapat dicegah apabila aplikasinya beragam, seperti program yang ditulis *user,* analis tidak mungkin untuk memiliki pengetahuan apapun tentang suatu aplikasi. Isu lainnya yang mungkin selama pengujian melibatkan penggunaan aplikasi yang terkait dengan kontrol keamanan, seperti enkripsi dan *password*. Banyak aplikasi yang digunakan seperti kontrol keamanan untuk mencegah akses yang tidak sah ke data yang sensitif oleh user yang diberi hak.

Dalam banyak kasus, analis perlu untuk membawa bersama data aplikasi yang bersangkutan dari beberapa sumber data aplikasi yang divariasikan; hal ini sebagian besar merupakan proses manual. Analisa yang terperinci dari rekonstruksi kejadian dan kejadian terkait biasanya diperlukan keahlian dan banyak pengetahuan yang dapat mengerti informasi yang dipresentasikan oleh semua sumber. Analis dapat meninjau ulang hasil dari pengujian sumber data aplikasi secara individu dan melihat bagaimana informasi yang sesuai bersama – sama. *Tool* mungkin dapat membantu analis mencakup perangkat lunak *security event management* (seperti yang diuraikan dalam bagian 6.2.5, yang mana bisa menghubungkan beberapa aplikasi yang berkaitan dengan kejadian antara banyaknya sumber data dan perangkat lunak untuk analisa *log* (meliputi beberapa jenis dari *host* terkait perangkat lunak yang mendeteksi gangguan), yang mana perlu dijalankan terhadap jenis *log* tertentu untuk mengidentifikasi kegiatan yang mencurigakan. Bagian 8 menyediakan contoh bagaimana data dari banyak jenis dari sumber yang dapat dikorelasikan melalui analisis untuk mendapatkan suatu pandangan yang lebih akurat dan menyeluruh dari apa yang terjadi.

## 7.5 Rekomendasi

Kunci rekomendasi yang dipresentasikan dalam bagian ini untuk menggunakan data dari aplikasi diringkas dibawah ini :

1. Analis perlu mempertimbangkan semua kemungkinan sumber data aplikasi. Kejadian aplikasi mungkin direkam oleh banyak sumber data yang berbeda. Juga, aplikasi mungkin digunakan melalui berbagai mekanisme, seperti berbagai program *client* yang diinstal pada suatu sistem dan Web yang terkait *client interfaces*. Dalam situasi yang sedemikian, analis perlu mengidentifikasi semua komponen aplikasi , menentukan yang mana dari kebanyakan yang mungkin menarik, menemukan lokasi masing – masing komponen yang penting dan memperoleh data.
2. Analis perlu membawa data aplikasi secara bersama – sama dari berbagai sumber. Analis harus meninjau ulang hasil dari pengujian sumber aplikasi individu dan menentukan bagaimana informasi yang sesuai bersama –sama, untuk melakukan analisis yang terperinci dari apikasi terkait kejadian dan rekonstruksi kejadian

# Penggunaan Data Dari Banyak sumber

# 8

Bagian 4 sampai 6 menguraikan perolehan dan pengujian data dari tiga kategori sumber data : file data, sistem operasi dan *network traffic*. Teknik dan proses untuk memperoleh dan menguji data dalam kategori ini pada dasarnya berbeda. Bagian 7 menguraikan perolehan dan pengujian data dari aplikasi data, yang mana membawa tiga kategori sumber data bersama. Sebagai contoh, banyak aplikasi yang menggunakan file data, merubah konfigurasi dari sistem operasi dan menghasilkan *network traffic*. Banyak situasi, seperti kejadian keamanan komputer, dapat ditangani secara efektif dengan meneliti berbagai tipe dari sumber data dan kejadian yang menghubungkan ke sumber lainnya.

Bagian ini menyangkut pemandu yang menghadirkan dua contoh bagaimana berbagai sumber data dapat digunakan bersama-sama selama suatu analisa. Masing-masing contoh menguraikan suatu skenario, menunjuk suatu kebutuhan spesifik untuk analisa data, dan menghadirkan suatu penjelasan bagaimana proses analisa dapat dilakukan. Penjelasan juga menggambarkan bagaimana kompleksnya proses analisa. Contohnya dihadirkan di dalam bagian ini sebagai berikut:

1. Menentukan *worm* mana yang telah menginfeksi suatu sistem dan mengidentifikasi karakteristik *worm*
2. Merekonstruksi urutan peristiwa *cyber* yang melibatkan *threatening e-mail*.

## 8.1 Layanan network yang dicurigai terinfeksi worm

Suatu *help desk* organisasi menerima beberapa panggilan dalam waktu yang singkat dari keluhan *user* tentang server tertentu yang menyediakan tanggapan yang lambat. *Help desk* mengirim inti masalah ke grup yang memonitor. Gangguan jaringan mereka terdeteksi sistem yang baru – baru ini melaporkan beberapa tanda yang tidak biasanya melibatkan server dan seorang analis yang meninjau ulang tanda tersebut percaya bahwa mereka mungkin akurat. Data dalam sinyal menunjukkan ada beberapa aktifitas yang mencurigakan yang dilangsungkan pada server dan server sekarang menghasilkan aktifitas indentik yang dilakukan pada sistem lainnya. Maka, gangguan yang dideteksi analis pada awal hipotesa

adalah *worm* yang mungkin telah menyerang layanan jaringan yang mudah diserang dan menginfeksi server, yang mana sekarang berusaha untuk menginfeksi sistem lainnya. Grup yang memonitor memanggil orang yang menangani kejadian untuk menyelidiki kejadian yang mungkin pada server tersebut.

Untuk suatu peristiwa, hal yang tertentu ini merupakan tugas *incident handler* untuk menentukan tipe dari *worm* yang telah menginfeksi sistem dan mengidentifikasi perbedaan karakteristik dari *worm*. Informasi ini kritis terhadap tim yang menanggapi kejadian tersebut, maka mereka dapat bertindak secara efektif untuk melakukan *containment, eradication* dan aktifitas pemulihan, seperti halnya pencegahan sistem lainnya dalam organisasi sejak terinfeksi. Jika investigasi incident handler memperlihatkan bahwa kejadian tersebut dapat disebabkan oleh sesuatu yang lain selain worm, maka identifikasi karakteristik oleh handler harus dapat membantu dalam menentukan apa yang sebenarnya terjadi.

Informasi mengenai kejadian ini mungkin direkam dalam beberapa tempat yang berbeda. *Incident handler* harus memastikan sumber data yang mungkin untuk memiliki informasi dulu, berdasar pada pengalaman *handler* dengan sumber data dan informasi awal yang tersedia sebelumnya mengenai peritiwa tersebut. Sebagai contohnya, karena jaringan sensor IDS melihat aktifitas yang mencurigakan, jaringan lainnya terkait dengan sumber data yang memonitor segmen jaringan yang sama mungkin juga berisi informasi yang relevan. Jika organisasi menggunakan *security event management* atau perangkat lunak *tool* analisis forensik jaringan, yang mana membawa data bersama – sama dari banyak sumber yang berbeda, *incident handler* mungkin mampu untuk mengumpulkan semua informasi yang penting hanya dengan menjalankan beberapa *query* dari *console* SEM atau NFAT. Jika sumber dari data yang dipusatkan tidak tersedia, hendler harus memeriksa setiap sumber yang berpotensi dari karakteristik serangannya, seperti yang berikut ini :

1. *IDS Network-Based*. Karena laporan awal dari suatu kejadian telah dihasilkan oleh jaringan sensor IDS, hal itu sangat mungkin dimana data jaringan IDS berisi informasi pada karakteristik dari aktifitas jaringan . Pada saat minimum, data harus menunjukkan server mana yang diserang dan pada nomor *port* apa, yang mana menunjukkan layanan jaringan mana yang ditargetkan. Mengidentifikasikan layanan sangatlah penting untuk mengidentifikasi sifat mudah sifat mudah diserang yang dimanfaatkan dan membangun strategi peringanan untuk mencegah kejadian yang sama dari yang terjadi pada sistem. Dari suatu sudut pandang analisa, mengetahui layanan dan nomor *port* yang ditarget juga

penting karena informasinya dapat digunakan untuk mengidentifikasikan sumber data lainnya yang mungkin dan *query* mereka untuk informasi yang relevan. Beberapa penyebaran jaringan IDS mungkin merekam informasi dengan manfaat tambahan, seperti data aplikasi (misalnya permintaan dan tanggapan Web , e-mail utama dan nama file *attachment*). Data aplikasi mungkin berisi kata – kata, frase atau urutan karakter lainnya yang dihubungkan dengan *worm* tertentu.

2. *Network-Based Firewall*. *Firewalls* secara khusus dikonfigurasi untuk mencatat usaha koneksi yang dihalangi, yang mana meliputi alamat dan *port* tujuan IP yang diharapkan. Maka, *firewalls* mungkin memiliki catatan mengenai aktifitas *worm* yang dihalangi mereka. Beberapa *worm* berusaha untuk memanfaatkan *multiple services* atau *service ports*; catatan *firewall* mungkin memperlihatkan *worm* yang mencoba secara nyata untuk membangun suatu koneksi dengan sedikitnya empat nomor *port* yang berbeda, tetapi koneksi blok *firewall* menggunakan tiga *port*. Informasi ini dapat berguna dalam mengidentifikasi *worm*. Jika *firewalls* dikonfigurasi untuk mencatat koneksi yang diijinkan, maka *log* mereka mungkin memperlihatkan *host* mana dalam organisasi yang menerima *worm traffic* atau telah diinfeksi atau dihasilkan *worm traffic* mereka sendiri. Hal ini berguna terutama untuk situasi dimana sensor jaringan IDS tidak mengawasi semua traffic yang menjangkau *firewalls* tersebut. Alat perimeter lainnya dimana mungkin telah dilewati *worm traffic*, seperti *routers*, VPN *gateways* dan remote *access servers*, mungkin catatan informasi yang serupa untuk dicatat oleh *network-based firewalls.*
3. *Host-Based IDS* dan *Firewall*. *Firewall* dan IDS berjalan pada sistem yang terinfeksi yang mungkin berisi banyak informasi yang detil dibanding produk *network-based IDS* and *firewall*. Sebagai contoh, host-based IDS dapat mengidentifikasi perubahan pada file dan *configuration settings* pada host yang dilakukan oleh *worm*. Informasi ini tidak hanya membantu dalam merencanakan penahanan, pemberantasan dan mengembalikan aktifitas dengan menentukan bagaimana *host* dipengaruhi *worm*, tetapi juga dalam mengidentifikasi *worm* mana yang menginfeksi sistem. Bagaimanapun, karena beberapa *worm* melumpuhkan *host* terkait kontrol keamanan dan *destroy log entries*, data dari *host-based IDS* dan perangkat lunak *firewall* mungkin dibatasi atau hilang. Jika perangkat lunak tersebut telah dikonfigurasi untuk meneruskan salinan dari *log* tersebut ke *log* server yang disentralisasi, maka *queries* untuk server tersebut mungkin menyediakan beberapa informasi.

4. Perangkat Lunak Antivirus. Sebab ancaman mencapai server dan sukses melanggarnya, hal itu tidak mungkin jaringan atau perangkat lunak *host-based* yang berisi catatan apapun tentangnya. Jika perangkat lunak antivirus telah mendeteksi *worm* , hal itu dapat menghentikannya. Bagaimanapun, hal ini mungkin dimana perangkat lunak antivirus melihat *worm* tetapi bagaimanapun juga gagal untuk menghentikannya atau perangkat lunak antivirus tersebut telah di-*update* sejak suatu infeksi dengan tanda yang baru yang dapat mengenali *worm*. *Incident handler* dapat juga meneliti server dari *worm* menggunakan perangkat lunak versi terbaru dari *toolkit* yang terpercaya.
5. Log aplikasi. Jika *worm* menggunakan protokol aplikasi yang biasa, seperti informasi HTTP atau SMTP mengenainya mungkin direkam dalam beberapa tempat, seperti aplikasi *server logs*, *proxy server* dan *application-specific security controls*. Lebih sedikit protokol aplikasi biasanya hanya mungkin memiliki informasi dalam aplikasi *server logs*. Log aplikasi mungkin merekam detil yang luas pada karakteristik *application-spesific* dari suatu aktifitas dan yang terutama sekali menolong saat mengidentifikasi karakteristik *application-spesific* dari lebih sedikit aplikasi umum.

Tujuan dalam informasi awal mengumpulkan usaha adalah untuk mengidentifikasi karakteristik yang cukup sedemikian sehingga *worm* dapat dikenal secara positif. Hal ini dapat menantang, terutama untuk *worm* yang memiliki lusinan jenis; jenis ini sering memiliki karakteristik yang serupa tetapi menyebabkan efek yang berbeda terhadap sistem.analis dapat melakukan *queries* pada basis data *malware* suatu *vendor* anti virus, mencari untuk mengenali karakteristik seperti nama produk, nama layanan atau nomor *port*, teks *string* didalam *malware* dan file atau *settings* yang dimodifikasi pada target. Hampir kejadian manapun dari *malware*, selain dari *lastest threats* (misalnya yang dirilis dalam beberapa jam yang lalu), harus dapat tercakup dalam basis data *malware* yang utama. Masing – masing *entry* basis data secara khusus berisi informasi yang luas pada tentang bagaimana *worm* tersebut menyebar, bagaimana hal tersebut mempengaruhi sistem (misalnya, perubahan apa yang dapat dibuatnya) dan bagaimana hal itu dapat dimusnahkan, termasuk ukuran untuk mencegah infeksi pada sistem lainnya.

Jika pencarian basis data *malware* tidak mengarahkan ke pengidentifikasian *worm*, maka *incident handler* mungkin perlu untuk melakukan analisis dan pencarian tambahan untuk menentukan informasi yang disediakan secara normal oleh *entry* basis data *malware*.

Meskipun suatu organisasi dapat mengirim salinan dari *worm* ke organisasi vendor anti virus untuk analisis dan identifikasi, organisasi harus melakukannya sendiri sejak *timeframe* untuk tanggapan suatu vendor tidak dikenal. Untuk mengumpulkan banyak infomasi, analis dapat menguji infeksi tersebut melalui dua cara sebagai berikut :

1. Status terbaru dari suatu *host.* Analis dapat melihat beberapa aspek berbeda dari status *host* yang sekarang. Dalam kasus ini, hal ini mungkin paling efektif untuk memulainya dengan menguji daftar koneksi suatu jaringan untuk mengidentifikasi koneksi yang tidak biasa(misalnya, sejumlah besar nomor, penggunaan nomor *port* yang tak diduga, *host* yang tak terduga) dan *listening ports* yang tak terduga (misalnya *backdoors* yang diciptakan oleh *worm*). Langkah lainnya yang mungkin dapat berguna terdiri dari mengidentifikasi proses yang tak dikenal didalam daftar proses yang sedang dijalankan dan menguji *host log* untuk mengungkapkan *entries* manapun yang tidak sesuai yang mungkin terkait terhadap infeksi tersebut.
2. Aktifitas jaringan *host*. Analis dapat mengumpulkan *worm traffic* yang sering dihasilkan dengan diinfeksikannya server melalui paket *sniffer* dan *protocol analyzer*. Hal ini mungkin menyediakan informasi tambahan yang cukup mengenai karakteristik dari *worm* yang mana analis dapat menemptkannya kedalam basis data malware yang utama.

Peristiwa *worm* sering mengharuskan secepat mungkin menanggapi kemungkinan, karena suatu sistem yang diinfeksi mungkin menyerang sistem lainnya di dalam atau diluar organisasi. Juga, *worm* sering menginstal *backdoors* dan *tool* lainnya ke sistem yang mengijinkan *attackers* untuk mendapatkan akses kendali untuk menginfeksi sistem, yang mana dapat mendorong ke arah kerusakan tambahan. Maka, organisasi mungkin memilih untuk dengan segera memutuskan hubungan sistem yang diinfeksi dari jaringan, sebagai ganti melakukan suatu analisis dari *host* yang dulu. Hal ini mungkin membuatnya sangat lebih sulit bagi analis untuk mengidentifikasi suatu *worm* dan menentukan efeknya pada sistem—sebagai contohnya, aktifitas jaringan dan aspek tertentu dari status *host* yang tidak tersedia. Analis mungkin perlu untuk melakukan analisa yang lebih detil dari server,seperti memperoleh sistem filenya dan mengujinya untuk tanda dari aktifitas kejahatan (misalnya *executables* sistem yang dirubah) untuk menentukan secara tepat apa yang terjadi terhadap server. Analis dapat juga menguji karakteristik *non-volatile* dari sistem operasi server, seperti mencari level administratif dari *account* grup dan user yang mungkin telah ditambahkan oleh

*worm*. Akhirnya, analis harus mengumpulkan informasi yang cukup untuk mengidentifikasi perilaku *worm* dalam detil yang jelas sedemikian sehingga *incident response team* dapat bertindak secara efektif untuk mengetahui, membasmi dan memulihkannya dari *incident.*

## 8.2 Mengancam E-mail

Tanggapan seorang *Incident Handler* terhadap permintaan untuk bantuan dengan suatu *malware incident.* Suatu sistem personil yang telah jadi terinfeksi, kelihatannya melalui suatu *e-mail* yang mengklaim telah dikirim dari *account e-mail* personil lainnya melalui sistem *e-mail* organisasi. *Incident handler* telah ditugaskan untuk menolong penginvestigasi dalam menemukan sumber data semua yang mungkin berisi catatan dari *e-mail*. Informasi ini akan berguna dalam menentukan sumber sebenarnya dari *email.* Karena *e-mail* dapat ditempa dengan mudah, hal ini penting untuk menggunakan semua sumber data untuk merekonstruksi urutan dari kejadian untuk membuat, mengirim dan menerima e-mail.

Bagian pertama dari informasi untuk meninjau ulang adalah *e-mail header.* Hal itu perlu berisi nama *domain* dan alamat IP dari *host* yang mengirim *e-mail,* jenis dari *e-mail* yang digunakan *client* untuk mengirim *e-mail, e-mail* dari pesan ID, dan tanggal dan waktu ketika *e-mail* dikirimkan. *E-mail header* harus juga mencatat masing – masing *server e-mail* (nama domain dan alamat IP) dimana pesan melewatinya, dan tanggal juga waktu masing – masing sistem memproses *e-mail.* Karena seperti yang diduga, *e-mail* dikirim dengan menggunakan sistem *e-mail* organisasi. Mengasumsi bahwa ini merupakan kasus, *incident handler* dapat mengecek masing –masing sistem pada daftar untuk mengkorelasikan informasi. Tergantung pada tipe dari *e-mail client* yang digunakan oleh *recipient* dan konfigurasinya, *e-mail* mungkin telah di*download* ke *recipient workstation* atau hal itu mungkin tetap berada pada server *e-mail*. Hal ini juga mungkin untuk *e-mail* tetap disimpan dalam kedua tempat.

Setelah meninjau ulang *header*, *incident handler* harus mengumpulkan lagi informasi pada pengiriman dari e-mail. *Header* harus mendata alamat IP dan *e-mai lclient* yang digunakan oleh pengirim;*incident handler* harus menentukan *host* mana yang menggunakan alamat IP pada waktu *e-mail* tersebut dikirim. Ada tiga kemungkinan untuk alamat IP yaitu sebagai berikut :

1. *E-mail client* lokal.

Dalam kasus tersebut, *incident handler* harus bisa menggunakan arsip jaringan, seperti *DHCP logs*, untuk mengidentifikasi desktop, laptop, PDA atau alat lainnya untuk mencari malware dan untuk merekam yang terkait dengan e-mail. Sebagai contoh, *e-mail client* mungkin telah dikonfigurasi untuk menyimpan salinan dari masing –masing e-mail yang dikirimnya atau *user* mungkin telah menyimpan *draft* dari pesan *e-mail*. Jika pesan tidak dapat ditemukan tetap utuh pada sistem, memperoleh data dari memori suatu alat dan sistem file, termasuk menghilangkan dan file sementara, mungkin mendorong ke arah identifikasi dari fragmen suatu *e-mail*. Juga kontrol keamanan pada alat seperti *spam filtering* dan perangkat lunak anti virus yang mungkin dapat mencari.

2. *E-mail* yang keluar dan mencatat arsipnya. Hal ini juga mungkin, tetapi mau tidak mau, salinan dari *e-mail* disimpan pada suatu server *e-mail*. Sebagai tambahan untuk mencari arsip dari *e-mail* pada *local host, incident handler* perlu juga menguji arsip otentikasi pada *host* untuk menentukan *user account* mana yang digunakan pada saat *e-mail* dikirim.
3. Server yang didasarkan *E-mail client*. Jika organisasi menawarkan server yang didasarkan pada *client*, seperti *Webbased e-mail interface*, maka alamat IP dapat sesuai dengan server itu. Secara khusus, penggunaan dari server memerlukan *user* untuk membuktikan keaslian mereka sendiri, jadi hal itu mungkin untuk mengotentikasi arsip yang ditunjuk ketika pengirim dituduh masuk ke dalam server dan alamat IP apa yang digunakan sistem *user*. Kemudian *incident handler* dapat menentukan sistem mana yang menugaskan alamat IP pada waktu tersebut dan menguji sistem yang diidentifikasi untuk *malware* dan *e-mail*. Sebagai contoh, file sementara dari *Web browser* dapat berisi salinan isi dari *e-mail*.
4. *Spoofed.* Jika IP alamat dibuat—sebagai contoh, hal itu bukan alamat yang valid didalam jaringan organisasi—kemudian *incident handler* perlu untuk bersandar pada sumber data lainnya untuk berusaha mengidentifikasi *host* yang sebenarnya mengirim pesan *e-mail.*

Server *e-mail* suatu organisasi adalah sumber lainnya yang mungkin dari informasi. Masing – masing server alamat IP didaftar dalam *e-mail header* yang harus berisi beberapa arsip dari *e-mail*, termasuk pesan nilai ID, yang mana perlu cepat memfasilitasi identifikasi dari arsip yang bersangkutan. Seperti yang disebutkan diawal, hal ini mungkin server *e-mail* yang terakhir dalam daftar berisi salinan dari *e-mail*. *Backups* dari server mungkin hanya berisi salinan dari *e-mail* jika hal itu telah dipegang untuk pengiriman untuk beberapa jam atau lebih. Layanan lainnya diasosiasikan dengan *e-mail*, seperti perangkat lunak anti virus

dan *spam filters*, mungkin juga berisi catatan dasar dari aktifitas *e-mail*, tetapi tidak mungkin untuk berisi banyak detil. Sumber mungkin lainnya dari informasi adalah arsip otentikasi. Meskipun beberapa server *e-mail* meminta *user* untuk otentikasi buat mengirim *e-mail*, mereka secara khusus melakukan otentikasi permintaan untuk pengiriman *e-mail* ke *users*. Karena *user* sering mengirim dan menerima *e-mail* selama sesi tunggal, *log* dari otentikasi mungkin berisi catatan untuk menerima *e-mail* yang dapat mebantu dalam menentukan siapa yang telah mengirimkan *e-mail* tertentu.

Sumber informasi lainnya yang mungkin adalah catatan dari *network traffic* yang dihasilkan dengan mengirimkan atau menerima *e-mail*. Paket *sniffer* atau *tool* untuk analisa forensik jaringan yang memonitor aktifitas jaringan mungkin telah menangkap suatu aktifitas, mencakup alamat IP yang sebenarnya dari pengiriman dan penerimaan *host*, muatan dan *header* dari e-mail dan aktifitas otentikasi manapun yang diasosiasikan.

Akhirnya, *incident handler* harus mengidentifikasi *host* yang digunakan untuk mengirim dan menerima *e-mail*, seperti halnya *intermediate host* yang memindahkan *e-mail* dari pengirim ke penerima. *Incident handler* harus mengumpulkan salinan dari *e-mail* dan mendukung informasi dari masing – masing *host* yang relevan dan menggunakan *timestamp* dalam merekam, membuat ulang susunan kejadian dari gambaraan *cyber*. Sebagai contoh, seorang *user* dicatat pada desktop komputer tertentu pada jam 20.37. pada jam 22.02 malam, *e-mail malware* telah dikirim dari komputer tersebut dengan menggunakan *built-in e-mail client*. *E-mail* melewati tiga dari *e-mail server* organisasi dan disimpan pada server 4 untuk menunggu perolehan kembali oleh *recipient* yang diharapkan.

*User* penerima masuk ke komputer laptop tertentu pada jam 23.20 dan men*download* e-mail jam 23.23, mencakup *e-mail malware*. Informasi ini kemudian dapat digunakan untuk mengidentifikasi dan memusnahkan semua salinan dari *e-mail malware*, seperti halnya menentukan *host* mana yang mungkin perlu untuk memiliki prosedur pemulihan tambahan yang dilakukan.

## 8.3 Rekomendasi

Kunci rekomendasi yang dipresentasikan dalam bagian ini untuk menggunakan data dari banyak sumber dirangkum seperti yang dibawah ini :

1. Analis dapat menangani banyak situasi secara sangat efektif dengan meneliti sumber data pribadi dan kemudian menghubungkan kejadian diantaranya. Proses dan teknik untuk

memperoleh dan menguji tipe yang berbeda dari sumber data yang pada dasarnya berbeda. Banyak aplikasi memilki data yang diambil dalam file data, sistem operasi dan *network traffic*.

2. Organisasi harus sadar akan teknis dan kompleksitas logistik dari analisis. Suatu kejadian tunggal dapat menghasilkan catatan pada banyak sumber data yang berbeda dan menghasilkan banyak informasi dibanding analis dapat kemungkinan meninjau ulang. *Tools* seperti SEM dapat membantu analis dengan membawa informasi bersama dari banyak sumber data dalam satu tempat.

# DAFTAR PUSTAKA


1. Abdul Kadir dan Terra Ch. Triwahyuni, Pengenalan Teknologi Informasi, Andi, Yogyakarta, 2003.

2. Frank J.Defler, Jr, Panduan Menggabungkan LAN, PT.Elex Media Komputindo, Jakarta,1992.

3. Bukti Digital, Kunci Penguak Kejahatan Cyber
www.pcmedia.co.id

4. Guide to Computer and Network Data Analysis: Applying Forensic Techniques to Incident Response
http://csrc.nist.gov/publications/nistpubs/index.html

5. Cybercrime and Intellectual Property
www.solusihukum.com

6. Seputar Cybercrime
http://www.cybertech.cbn.net.id

7. Cybercrime Act 2001
www.findlaw.com.au

8. Firewall Pada Linux
www.ristishop.com