Forensik Jaringan & Cloud

Bahasan Meliputi :
Network Forensics | Computer Network
Internetworking | Computer Security
Network Forensics Tools | Network Forensics Akuisisi
Wireshark | Case Study | Cloud Forensics

Imam Riadi | Ade Kurniawan

Forensik Jaringan dan Cloud

Imam Riadi | Ade Kurniawan

deandra kreatif

Forensik Jaringan dan Cloud

Penulis: Imam Riadi dan Ade Kurniawan
Proof: Diandra Kreatif
Layout: Diandra Kreatif
Cover: Diandra Kreatif

Diterbitkan melalui:
Diandra Kreatif (Kelompok Penerbit Diandra)
Jl. Melati No. 171
Sambilegi Baru Kidul, Maguwoharjo, Depok, Sleman, Yogyakarta
Telp. (0274) 2801996, Fax. (0274) 485222
E-mail: diandracreative@gmail.com
Fb. DiandraCreative SelfPublishing dan Percetakan
Instagram: diandraredaksi, diandracreative
www.diandracreative.com

Cetakan 2, Februari 2020
Yogyakarta, Diandra Kreatif 2020
xiv + 179 hlm; 15,5 x 23 cm
ISBN:

Prakata

Masyarakat menggunakan komputer, laptop, dan *smartphone* setiap hari dan berkomunikasi melalui internet untuk mengakses berbagai aplikasi karena terhubung adanya jaringan komputer. Saat ini, persentase pertumbuhan jumlah kejahatan *cyber* dan serangan *malware* baik melewati atau menyerang perangkat jaringan dan *user* telah berada pada titik mengkhawatirkan. Di lingkungan *network computer, information security*, dan *digital forensics*. *Network forensics* merupakan suatu bidang studi baru yang berkembang begitu sangat pesat. *Network forensics* atau dikenal juga dengan nama *forensika jaringan* umumnya mengacu pada studi ilmiah tentang bukti berbasis jaringan yang dikembangkan untuk tujuan penyelidikan hukum.

Dalam buku ini, kami berusaha memberikan landasan teoretis, teknis, hukum, dan juga disertai studi kasus. Buku ini ditujukan untuk khalayak akademis dan profesional. Sebagai buku teks, dimaksudkan sebagai buku ajar di tingkat sarjana atau pascasarjana di bidang ilmu komputer, teknologi informasi, keamanan jaringan, dan forensika digital. Buku ini berfungsi sebagai referensi dasar bagi para peneliti di

forensika jaringan, dan akan berguna bagi praktisi dan penyidik digital forensik.

Sebagai penutup.

Buku ini jauh dari sempurna, saran dan pertanyaan dapat dilayangkan ke: imam.riadi@mti.uad.ac.id atau ade.kurniawan@uvers.ac.id

Salam Hangat,

Penulis

Daftar Isi

Daftar Gambar:

Bab 1
Network Forensic

Tujuan Instruksional Umum:

1. Mahasiswa/wi mampu menjelaskan tentang *network forensics*.

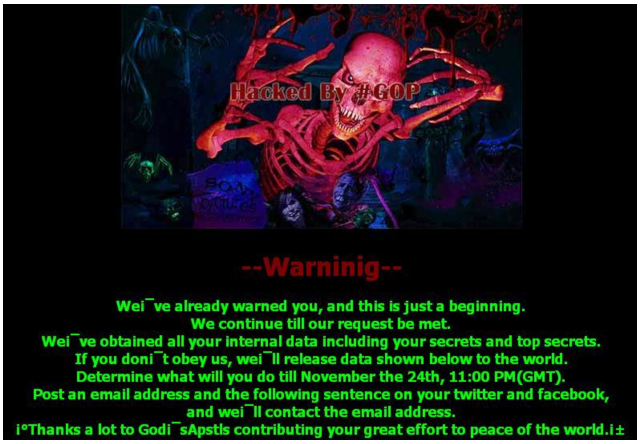
Tujuan Instruksional Khusus:

1. Mahasiswa/wi mampu menjelaskan istilah standar dalam *network forensics*.
2. Mahasiswa/wi mampu memahami latar belakang, definisi, aplikasinya, dan klasifikasi dalam *network forensics*.
3. Mahasiswa/wi dapat menjelajahi bidang penelitian *network forensics* yang sedang berkembang.

1.1 Pendahuluan

Pada tanggal 24 November 2014, *Los Angeles Times* melaporkan bahwa karyawan studio Sony Pictures Entertainment dari Culver City mendapatkan pesan dengan peringatan mengancam dengan gambar kerangka merah yang muncul di layar ketika karyawan mencoba masuk ke komputer kerja mereka, seperti terlihat pada Gambar 1.1. Tersiar

ke publik di seluruh dunia, Sony Pictures Entertainment telah diretas, informasi pribadi dan informasi karyawan tentang tanggungan mereka, komunikasi *email* antara karyawan, informasi gaji eksekutif, dan salinan film Sony yang akan dirilis ke publik telah dicuri oleh pihak peretas.



Gambar 1.1 Sony: Hacked By GO

Direktur FBI James Comey mengatakan sekelompok *hacker* yang menamakan dirinya "*Guardian of Peace*/GOP" dituduh oleh pemerintah Amerika Serikat atas serangan terhadap Sony Pictures Entertainment. Serangan ke Sony tersebut telah meninggalkan petunjuk yang mengarah pada keterlibatan pemerintah Korea Utara. Kelompok tersebut sebelumnya mengirim *e-mail* yang mengancam ke Sony, menggunakan alamat penyedia internet yang digunakan secara eksklusif di pakai oleh Korea Utara. Direktur FBI James Comey mengatakan bahwa banyak pakar keamanan yang tidak menyetujui pandangannya karena menganggap FBI tidak memiliki fakta dan bukti yang cukup. Berbagai pakar keamanan memeriksa bukti yang ditinggalkan oleh penyerang, dan penelitian mereka memberikan wawasan tentang sumber serangan ini. Meski tidak definitif, analisis mereka memberikan gambaran yang

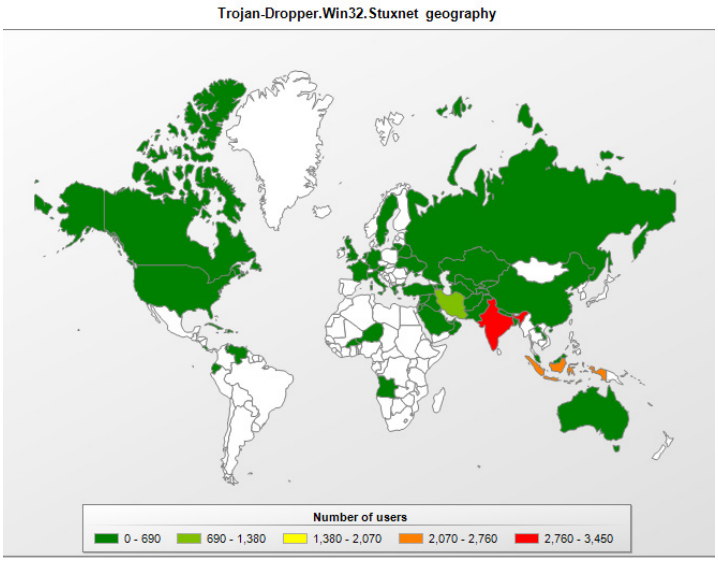
jauh lebih jelas dan menyarankan sebuah kelompok kriminal terorganisir yang beroperasi di Romania bertanggung jawab atas serangan Sony Pictures Entertainment tepat di hari Senin, 24 November 2014.

Dampak serangan terhadap Sony Pictures Entertainment, akhirnya para ahli mampu merekonstruksi serangan dari bawah hingga ke atas dan menemukan sejumlah alamat IP yang terkait dengan serangan lain yang juga dikaitkan dengan aktor di Rumania. Kehadiran teks Rumania dalam *string malware* ditemukan selama penyelidikan forensik. Mereka menemukan bahwa *malware* dikirim menggunakan teknik *"spear phishing"* yang ditargetkan pada eksekutif tingkat atas Sony Pictures Entertainment pada tanggal 13 November 2014. Sehari kemudian, *malware* mulai berkomunikasi dengan server *Command Control* (C2) dan menyebar dengan menggunakan *Server Message Block* (SMB). Sembilan hari kemudian, sebuah akun bernama *Guardians Of Peace* masuk ke *Pastebin* dan merilis data rahasia Sony ke publik.

Pada tanggal 25 April 2011, Iran telah diserang oleh *worm* komputer baru bernama Stars yang menyebabkan kerusakan minimal pada tahap awal, dan *worm* tersebut kemungkinan *file executable*. Stars adalah *worm* komputer kedua yang menargetkan Iran setelah *worm* Stuxnet, yang mampu mengambil alih pembangkit listrik dan telah menginfeksi banyak lokasi industri di seluruh dunia. *Worm* W32. Stuxnet telah menjadi fokus media dan peneliti. Stuxnet ditemukan pada bulan Juni/Juli 2010 dan merupakan salah satu ancaman kompleks kerena menargetkan sistem kontrol industri dan memodifikasi kode pada *programmable logic controllers* (PLC) yang dapat diprogram.

Stuxnet menggunakan teknik penghindaran antivirus, kode injeksi proses kompleks, empat kerentanan *zero-day* yang terpisah, dan *rootkit* pertama yang dirancang khusus untuk sistem PLC. Penyebaran melalui sistem operasi berbasis Windows dan perangkat *universal serial bus* (USB), memasukan *rootkit* untuk bersembunyi, berkompromi

dengan administrator, dan menggunakan sertifikat digital palsu untuk menjadikan dirinya sebagai perangkat lunak tepercaya. Stuxnet menargetkan PLC berbassi Siemens SIMATIC WinCC atau STEP 7 *supervisory control and data acquisition systems* (SCADA). *Worm* komputer Stuxnet mungkin dirancang khusus untuk menyerang program nuklir Iran karena telah menginfiltrasi sistem industri di Iran dan berpotensi melumpuhkan sistem pendingin instalasi nuklir Iran. Terlihat pada Gambar 1.2, Sebaran negara-negar yang terinfeksi dengan *Rootkit*. Win32.Stuxnet.



Gambar 1.2 Sebaran Negara yang Terinfeksi dengan Stuxnet.

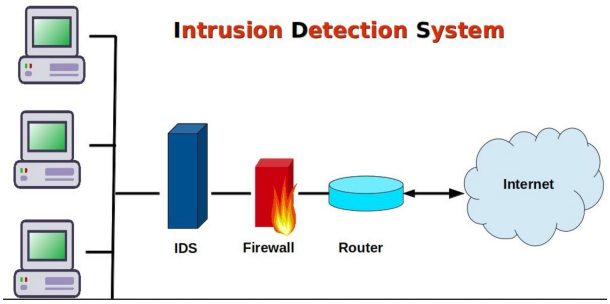
Stuxnet mengeksploitasi banyak kerentanan Windows dan setidaknya empat di antaranya adalah kerentanan *zero-day* (MS08-067 RPC Exploit, MS10-046 Eksploitasi LNK, MS10-061 Spool Server Exploit, MS10-073 Win32k.sys Exploit, MS10-092). Pejabat keamanan Iran yang menangani Stuxnet menunjukkan bahwa ancaman tersebut belum sepenuhnya dihapuskan karena *worm* dapat memiliki

siklus hidup yang tetap aktif dalam aktivitas mereka dalam bentuk lain. Mereka juga menyoroti bahwa Iran harus bersiap menghadapi tantangan *worm* di masa depan, yang dapat menginfeksi infrastruktur negara tersebut. Stuxnet telah membawa komunitas keamanan, sebuah kemungkinan yang mencolok dari ancaman serius terhadap kedaulatan negara di mana pun. *Network forensics* jelas merupakan salah satu cara untuk bersiap menghadapi kejadian seperti itu.

Infosecurity melaporkan bahwa ada peningkatan permintaan untuk *network forensics* karena perusahaan ingin memastikan siapa, bagaimana, kapan, mengapa, di mana, dan bagaimana layanan mereka diakses dan digunakan. *Network forensics* tidak bisa menghentikan serangan seperti Stuxnet, namun bisa memberikan cara untuk mengurangi dampaknya dengan memberikan analisis yang memungkinkan respons lebih cepat terhadap infeksi. Solera Networks menjelaskan bahwa *network forensics* mempersiapkan organisasi untuk merespons dengan cepat dan menemukan ancaman yang tidak diketahui. Ini meningkatkan nilai dan efektivitas investasi keamanan lainnya. *Network forensics* mengurangi dan menyederhanakan waktu pemantauan, pelaporan, analisis, dan remediasi yang dibutuhkan untuk mempertahankan diri dari serangan. Ini membantu penuntutan melalui bukti yang secara forensik lengkap dan memberikan pemahaman tentang akar penyebab pelanggaran keamanan untuk memungkinkan respons yang cepat, cerdas, dan efektif untuk mencegah kejadian bencana dan risiko berkelanjutan. Hal ini memungkinkan untuk perbaikan setelah pelanggaran terjadi melalui kemampuan untuk memutar ulang serangan jaringan.

Network forensics/forensik jaringan tampaknya serupa dengan *network security*/keamanan jaringan. Namun tujuan keduanya sangat berbeda. *Network forensics* adalah ilmu yang baru lahir yang berhubungan dengan penangkapan, rekam, dan analisis lalu lintas jaringan. Data lalu lintas jaringan ditangkap menggunakan *packet sniffers, alert* dan log

dikumpulkan dari alat keamanan jaringan yang ada. Data ini dianalisis untuk karakterisasi serangan dan diselidiki untuk melacak kembali pelanggarnya. Pendekatan keamanan jaringan menggunakan mekanisme defensif seperti *firewall* dan *intrusion detection system* (IDS). *Firewall* digunakan untuk pencegahan dan IDS untuk deteksi. Pendekatan ini secara stereotip menemukan kerentanan jaringan dan memblokir semua komunikasi berbahaya dari luar. *Firewall* mengendalikan lalu lintas yang memasuki jaringan dan meninggalkan jaringan, berdasarkan alamat sumber dan tujuan dan nomor *port*. Menyaring lalu lintas jaringan berbahaya sesuai aturan *firewall*. Sulit untuk memperbarui *intrusion signatures* karena semua kerentanan-kerentanan baru akan selalu terjadi.



Gambar 1.3 Intrusion Detection System (IDS)

Intrusion detection system (IDS) berfungsi untuk belajar, mendeteksi, dan melaporkan serangan saat terjadi secara *real time* dan tidak memiliki fitur pengumpulan bukti. IDS terdiri dari dua jenis: *signature-based (misuse) detection* dan *statistical-based (anomaly) detection*. Pencocokan pola dilakukan pada IDS *signature-based* untuk mendeteksi intrusi "*signature*", *IDS* tidak bisa mendeteksi serangan baru namun memiliki tingkat *false positive* rendah. IDS berbasis anomali/*statistical-based (anomaly) detection* melakukan pemantauan aktivitas dan mampu mendeteksi serangan baru namun memiliki tingkat *false positive* yang

lebih tinggi. Pendekatan *network forensics* mampu mengumpulkan bukti yang diperlukan untuk menanggapi insiden dan penyelidikan kejahatan. Sedangkan keamanan jaringan hanya melindungi sistem dari serangan. Alat keamanan jaringan seperti IDS yang terlihat pada Gambar 1.3 terus-menerus memantau untuk memonitor kemungkinan perilaku berbahaya. *Network forensics* mencakup penyelidikan *postmortem* atas serangan tersebut. Ini adalah kasus yang spesifik karena setiap skenario kejahatan berbeda dalam banyak aspek, dan terikat dengan proses waktu, mungkin juga ada kejahatan tertentu yang tidak melanggar kebijakan keamanan jaringan namun mungkin secara hukum dapat diterima. Kejahatan ini hanya dapat ditangani oleh *network forensics*. Perbedaan utama antara keamanan jaringan dan *network forensics* diberikan pada Tabel 1.1. sedangkan pada Tabel 1.2 akan dijelaskan perbandingan *computer forensics* dengan *network forensics.*

Tabel.1 1 Perbandingan *Network Forensics* dengan *Network Security*

Network Security	**Network Forensics**
Sistem proteksi terhadap serangan	Tidak ada proteksi sistem terhadap serangan
Biasanya secara ***real time***	*Postmortem*
Generalized - mencari kemungkinan perilaku berbahaya	Kasus dibatasi - ingin merekonstruksi skenario kriminal
Tetap waspada 24 jam setiap hari	Menunggu laporan terjadi tindakan kriminal
Bidang ilmu komputer yang telah matang atau mapan	Bidang ilmu baru belum matang

Tabel.1 2 Perbandingan *Computer Forensics* dengan *Network Forensics*

Computer Forensics	**Network Forensics**
Diperkenalkan oleh penegak hukum untuk menangani data komputer	Sebagai respons terhadap komunitas *hacker*
Penyelidik dan penyerang berada di dua tingkat yang berbeda	Penyelidik dan penyerang berada pada level keterampilan yang sama
Penyelidik dan penyerang menggunakan alat yang berbeda	Penyelidik dan penyerang menggunakan alat yang berbeda
Computer forensics berisi pelestarian, identifikasi, ekstraksi, dokumentasi, dan interpretasi data	*Network forensics* melibatkan penangkapan, rekam, dan analisis kejadian jaringan.

Network forensics dapat didefinisikan secara umum sebagai ilmu untuk menemukan dan mengambil informasi di jaringan menjadikanya sebagai bukti digital (*evidence*) dari suatu tindak kejahatan sehingga dapat diterima di pengadilan. Investigasi kejahatan *cyber* sering melibatkan kasus-kasus yang berkaitan dengan keamanan dalam negeri, spionase perusahaan, pornografi anak, kejahatan tradisional yang dibantu oleh teknologi komputer dan jaringan, pemantauan karyawan, atau catatan medis, di mana privasi memegang peran penting.

Network forensics adalah perpanjangan tangan dari *computer forensics*. *Computer forensics* diperkenalkan oleh penegak hukum dan memiliki banyak asas panduan dari metodologi investigasi sistem

peradilan. *Computer forensics* melibatkan pelestarian, identifikasi, ekstraksi, dokumentasi, dan interpretasi data komputer. *Network forensics* berevolusi sebagai respons terhadap komunitas *hacker* dan melibatkan penangkapan, perekaman, dan analisis kejadian jaringan untuk mengetahui sumber serangan.

Penyidik dan peretas di dalam *computer forensics* berada pada tingkat keahlian yang berbeda, sedangkan di *network forensics*, penyidik dan penyerang berada pada tingkat keterampilan yang sama. Peretas menggunakan seperangkat alat untuk meluncurkan serangan, dan di sisi yang lain penyidik *network forensics* menggunakan alat serupa untuk menyelidiki serangan tersebut. Peretas memiliki banyak waktu untuk meningkatkan keterampilan, dan ditambah dimotivasi pribadi, komunitas, negara atau oleh iming-iming keuntungan finasial.

Network forensics didefinisikan sebagai penggunaan teknik yang dapat dibuktikan secara ilmiah untuk mengumpulkan, memadukan, mengidentifikasi, memeriksa, mengorelasikan, menganalisis, dan mendokumentasikan bukti digital dari banyak sumber bukti digital yang berbasis jaringan. Definisi singkat dari *network forensics* adalah penangkapan, perekaman, dan analisis kejadian jaringan untuk mengetahui sumber serangan keamanan atau insiden.

1.2 Klasifikasi *Network Forensics*

Network forensics terdiri dari pemantauan lalu lintas jaringan dan menentukan apakah ada anomali di lalu lintas dan memastikan apakah itu mengindikasikan adanya serangan. Jika serangan terdeteksi, selanjutnya menentukan sifat serangannya. Teknik *network forensics* memberdayakan penyidik untuk melacak kembali ke penyerang dan memberikan bukti yang cukup untuk memungkinkan pelaku diadili sebagai tujuan utamanya.

Karakteristik sistem *network forensics* diklasifikasikan ke dalam tipe yang berbeda, di antaranya berdasarkan berbagai:

- ***Purpose***/tujuan *general network forensics* (GNF): berfokus pada peningkatan keamanan. Data lalu lintas jaringan dianalisis, dan pola serangan ditemukan. *Network forensics strict network forensics* (SNF) melibatkan persyaratan hukum yang ketat karena hasil yang diperoleh akan digunakan sebagai bukti untuk penuntutan kejahatan jaringan.
- **Paket *capture*/**sistem *capture packet*: menangkap semua paket yang melewati titik lalu lintas tertentu dan kemudian menganalisanya, membutuhkan penyimpanan dalam jumlah besar. Sistem *stop-look-and-listen* menganalisis setiap paket dalam memori, dan hanya informasi tertentu yang disimpan untuk analisis masa depan, yang membutuhkan prosesor yang lebih cepat.
- **Platform**: sistem *network forensics* dalam satu perangkat keras dan atau perangkat lunak. Ini bisa menangkap data, menganalisisnya dan mempresentasikan hasilnya di *interface* komputer. Bisa juga *software standalone*, yang bisa dipasang di *host*. Ini menganalisis tangkapan paket atau catatan NetFlow, yang disalin dan disimpan di *host*.
- ***Time of analysis***: perangkat komersial analisis di *network forensics* yang melibatkan surveilans jaringan, *signature-based anomaly detection*, analisis data, dan penyelidikan forensik. Banyak perangkat lunak sumber terbuka dirancang untuk penyelidikan *post-mortem* tentang paket *capture*. Data paket lengkap di-*capture*/ditangkap oleh alat sniffer, disimpan di *host* dan dianalisis secara *offline* di lain waktu.

- **Data *source***: Sistem *flow* berbasis sumber data mengumpulkan informasi statistik berdasarkan beberapa kriteria dalam lalu lintas jaringan saat melewati jaringan.

Network forensics memastikan bahwa penyerang menghabiskan lebih banyak waktu dan energi untuk menutupi jejaknya, sehingga membuat usaha serangan menjadi mahal. Penyerang di jaringan akan lebih berhati-hati untuk menghindari tuntutan atas tindakan ilegal mereka. Ini bertindak sebagai pencegah dan dapat mengurangi tingkat kejahatan jaringan, sehingga meningkatkan keamanan. Banyaknya insiden keamanan yang memengaruhi banyak organisasi dan meningkatnya kecanggihan serangan *cyber* adalah kekuatan pendorong utama di balik *network forensics*. Penyerang yang sukses sering memastikan bahwa mereka menutupi jejak mereka. Serangan yang tidak berhasil sering tidak diketahui, dan sedikit informasi tersedia untuk membantu diagnosis bahkan ketika mereka memperhatikannya.

Penyedia layanan internet (ISP) juga bertanggung jawab atas apa yang melewati jaringan mereka. Perusahaan yang melakukan bisnis di internet tidak dapat menyembunyikan pelanggaran keamanan dan sekarang diharapkan dapat membuktikan keadaan keamanan mereka sebagai tindakan kepatuhan untuk tujuan peraturan. Standar ISO 27001/27002 tentang teknologi informasi - teknik keamanan - manajemen keamanan informasi menetapkan persyaratan untuk menetapkan, menerapkan, mengoperasikan, memantau, meninjau, memelihara, dan memperbaiki sistem manajemen keamanan informasi terdokumentasi (*information security management system:ISMS)*. *Network forensics* juga memfasilitasi pencatatan bukti untuk penyelidikan dan membantu dalam memahami metodologi penyerang dan memberikan wawasan tentang alat yang digunakan oleh penyerang dan menemukan cara pertahanan baru. *Network forensics* juga bisa memberikan informasi

dan menjelaskan kekurangan pada alat keamanan jaringan telah terpasang.

1.3 Tren Penelitian dalam *Network Forensics*

Network forensics secara tradisional diterapkan lingkungan berkabel dan difokuskan pada protokol di lapisan protokol TCP/IP. Berikut adalah beberapa penelitian terbaru di bidang *network* forensics:

1. Penyerang menggunakan teknik steganografi yang untuk memberi keluesan pada *rootkit* untuk bersembunyi dan menghindari deteksi agar pola serangan menjadi lebih sulit, yang jika tidak mudah ditemukan oleh IDS.
2. *Honeypot forensics* ditempatkan untuk memberikan information teknik dan alat yang digunakan oleh peretas, sebelum dan sesudah penyerangan agar bisa menemukan bentuk, pola, alat dan teknik serangan baru dari *rootkit*, trojan, dan potensi eksploitasi *zero-day*.
3. Forensik internet IPv6: IPv6 menyediakan kepada penyerang berbahaya sebuah tempat berlindung sementara karena IPv6 cukup sulit untuk dipantau. Banyak *tunneling* jaringan gratis menyediakan konektivitas sederhana dan relatif anonim.
4. *Botnet forensics*: sebuah atau beberapa *robot* di *net* yang singkat menjadi ke bentuk kata *botnet* yang berfungsi untuk dikompromikan dapat dihubungkan ke sebuah jaringan di bawah kendali eksternal yang di sebut *bootmaster*, yang digunakan untuk mengirim *e-mail* spam atau menonaktifkan situs web dengan cara permintaan palsu yang banyak atau sering dikenal dengan DDoS. Sangat sulit untuk melacak identitas *spammer* dengan hanya menganalisis jejak elektronik.

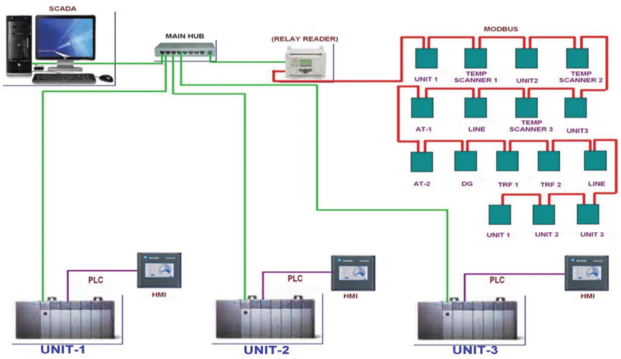
5. *Wireless network forensics*: teknologi nirkabel seperti terlihat pada Gambar 1.4 berkembang dengan cepat, frekuensi kebocoran data dan pencurian terus meningkat. Ada kebutuhan besar kekurangan alat dan prosedur untuk penyelidikan komputasi forensik yang menangani perangkat nirkabel secara efektif. Jaringan VoIP *over wireless* (VoIPoW) menjadi sistem komunikasi *mobile* yang paling populer di dunia. Namun, studi tentang serangan terhadap *wireless* jaringan VoIP masih dalam tahap awal. Tantangan ada di *mobile ad hoc networks* (MANETs) di mana jumlah paket bukti dikendalikan oleh tingkat reliabilitas dari jaringannya.

Gambar 1.4 *Wireless Acces Point*

6. *Application layer forensics* telah berpindah dari lapisan *network* dan *transport* ke lapisan *application* dari paket protokol TCP/IP. Serangan terhadap keamanan web mencakup *cross site scripting* (XSS), injeksi SQL, *buffer overflows* dan lain-lain. Bukti digital yang dapat diandalkan dapat diberikan berupa muatan lalu

lintas data jaringan yang dikirim ke dan dari layanan web. Forensik DNS juga merupakan tantangan penting.

7. *Supervisory control and data acquisition systems* (SCADA) *network forensics* banyak digunakan dalam pengendalian industri dan otomasi. Protokol SCADA modern seperti terlihat pada Gambar 1.5 sering menggunakan TCP/IP untuk men-*transport* data sensor dan sinyal kontrol.



Gambar 1.5 Scada: *Supervisory Control and Data Acquisition Systems*

Penggunaan TCP/IP sebagai protokol *carrier* dan interkoneksi jaringan IT dan SCADA menimbulkan masalah keamanan yang serius. Serangan yang berhasil pada jaringan TI dan perangkat jaringan *gateway* SCADA bisa menimbulkan malapetaka pada proses industri.

8. Grid forensics: komputasi *grid* mengumpulkan semua jenis sumber daya heterogen yang didistribusikan secara geografis dan memerlukan layanan keamanan mendalam untuk melindungi sumber daya dan data. Memerlukan teknik forensik yang sesuai dan dapat digunakan untuk menilai tanggung jawab para pelaku.

9. *Cloud computing: cloud computing forensics* akan memerlukan perubahan dalam kebijakan perusahaan dan keamanan terkait akses jarak jauh, penggunaan data melalui *browser*, mekanisme privasi dan audit, sistem pelaporan, dan sistem manajemen yang menggabungkan bagaimana data diamankan pada sistem *cloud computing* itu sendiri. Kompleksitas keterkaitan antara penyedia *cloud* dan konsumen berdamapak menjadi lahan subur bagi peretas dan penjahat. *Network forensics* dalam *cloud computing* memerlukan pola pikir investigasi baru, di mana beberapa data tidak tersedia, beberapa data akan dicurigai, dan hanya beberapa data yang akan disiapkan ke pengadilan.
10. *Network forensics* sistem cerdas: *network forensics* memerlukan pengetahuan tentang *intrusions signatures* untuk merancang skenario intrusi, bukti, dampak, tujuan intrusi dan membuat atribusi serangan. Pengetahuan pemecahan masalah menggambarkan bagaimana sistem dapat menggunakan pengetahuan domain untuk menganalisis aktivitas berbahaya menjadi sangat penting untuk merancang metode ontologi analisis *network forensics*.

Pelestarian bukti digital pada lalu lintas jaringan sangat tidak stabil (dinamis) dan harus di-*capture* dan disimpan dengan segera, jika tidak maka bukti digital tersebut akan hilang selamanya. Sebagian besar alat keamanan jaringan tidak menghasilkan nilai *hash* untuk data yang diambil atau menggunakan *algoritma hash* yang sama sehingga menghasilkan inkonsistensi. Integritas data yang terkumpul harus dipelihara sehingga data yang ditangkap akan melewati prosedur hukum yang ketat dan memenuhi syarat sebagai bukti di pengadilan.

Packet capture dari lalu lintas jaringan secara *real-time* ditransmisikan ke seluruh jaringan dengan kecepatan tinggi bisa berdampak paket

yang hilang dan menjadi tantangan penting. *Capture* paket penuh akan menghasilkan jumlah data yang sangat besar, prosesnya bisa dibuat efisien dengan mengumpulkan data yang berguna saja. Data yang dikumpulkan dapat dikurangi dengan memfilter data sesuai aturan yang disesuaikan untuk tujuan tertentu. Perangkat keamanan jaringan harus dapat menangani format masukan yang unik dan menghasilkan format *output* yang berbeda dan juga harus memfasilitasi sinkronisasi waktu tampilan universal dalam format yang berbeda dan zona waktu yang bervariasi di antara perangkat.

Metode *traceback* IP dapat melacak paket anonim internet kembali ke sumber asal serangan. Metode ini tidak bergantung pada pengetahuan atau kerja sama dan intervensi ISP. Penyerang dapat memulai serangan dalam waktu yang sangat singkat dan hanya menggunakan beberapa paket yang membuat proses *trace back* sulit.

1.4 Soal Latihan

1) Jelaskan apa saja yang melatar belakangi hadirnya *network forensics*!
2) Jelaskan perbedaan antara *network forensics* dengan *network security* dan *network forensics* dengan *computer forensics*!
3) Jelaskan secara singkat klasifikasi karakteristik dari sistem *network forensics*!
4) Jelaskan secara singkat tren-tren yang sedang berkembang dalam *network forensics*!

Bab 2
Computer Network

Tujuan Instruksional Umum:

1. Mahasiswa/wi mampu menjelaskan tentang sejarah jaringan komputer dan internet.
2. Mahasiswa/wi mampu menjelaskan tentang prinsip-prinsip dasar *internetworking*.

Tujuan Instruksional Khusus:

1. Mahasiswa/wi mampu memahami pengembangan *packet switching*.
2. Mahasiswa/wi mampu memahami latar belakang jaringan internet di era: 1972-1980, 1980-1990, Ledakan Internet: 1990-an, dan cara kerja dari internet.
3. Mahasiswa/wi mampu memahami apa itu dan cara kerja dari: *protocol, model open systems interconnection (OSI), internet protocolsuite (TCP/IP), internet protocol (IP),* dan *transmission control protocol (TCP)*.

2.1 Pendahuluan

Setiap satu abad peradaban manusia diakhiri oleh satu inovasi teknologi baru. Dilihat dari tiga abad terahir, dimulai dari abad ke-18 yang disebut era sistem mekanis. Menjelang abad ke-19, hadir sebuah inovasi mesin uap yang dikenal dengan Revolusi Industri. Selama abad ke-20 yang dikenal dengan era digital, di mana teknologi memegang peranan kunci peradaban manusia di dalam pengumpulan, pengolahan, dan pendistribusian informasi. Pemasangan jaringan telepon di seluruh dunia, penemuan radio dan televisi, kelahiran dan pertumbuhan industri komputer yang belum pernah terjadi sebelumnya, peluncuran satelit komunikasi, dan tentu saja internet yang telah banyak mengubah wajah dunia saat ini.

Kemampuan untuk mengumpulkan, memproses, dan mendistribusikan informasi, tumbuh signifikan atas permintaan akan pemrosesan informasi yang semakin canggih dan cepat. Organisasi dengan ratusan kantor tersebar di wilayah geografis yang luas berharap dapat memeriksa dan mengirimkan laporan status terkini dari setiap cabang organisasinya hanya dengan menekan satu tombol.

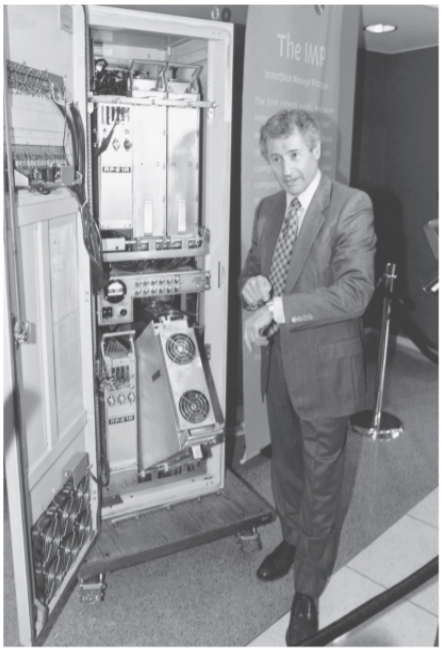
2.2 Sejarah Jaringan Komputer dan Internet

2.2.1 Pengembangan Packet Switching: 1961 -1972

Bidang jaringan komputer dan internet saat ini berawal di tahun 1960-an, ketika jaringan telepon saat itu merupakan jaringan komunikasi utama atau dominan di dunia. Di saat itu, jaringan telepon menggunakan *circuit switching* untuk mengirimkan informasi dari pengirim ke penerima, *signal* analog suara ditransmisikan pada tingkat konstan antara pengirim dan penerima. Mengingat semakin pentingnya komputer, pada awal tahun 1960-an muncul komputer *timeshared* yang berfungsi untuk menghubungkan komputer secara bersama-sama sehingga bisa dibagi di antara pengguna yang terdistribusi secara

geografis. Dampak dari lalu lintas yang dihasilkan oleh pengguna begitu besar seperti pengiriman perintah dari ke komputer yang letak geografisnya sangat jauh dan diikuti oleh periode tidak aktif sambil menunggu jawaban atau mempertimbangkan tanggapan yang diterima berdampak kepada keandalan dari jaringan saat itu. Oleh karena itu, solusi yang ditawarkan adalah menggunakan *packet-switching* sebagai alternatif yang efisien dan tangguh untuk rangkaian *switching*.

Kleinrock pada tahun 1964, merupakan seorang mahasiswa pascasarjana di MIT membuat suatu *packet-switching* dengan menggunakan teori antrean. Inovasi Kleinrock menunjukkan keefektifan pendekatan *packet-switching* untuk manajemen sumber daya lalu lintas bergitu besar.



Gambar 2.1 *Packet Switching* Pertama di Dunia di Tahun 1969
(Tanenbaum, 1996)

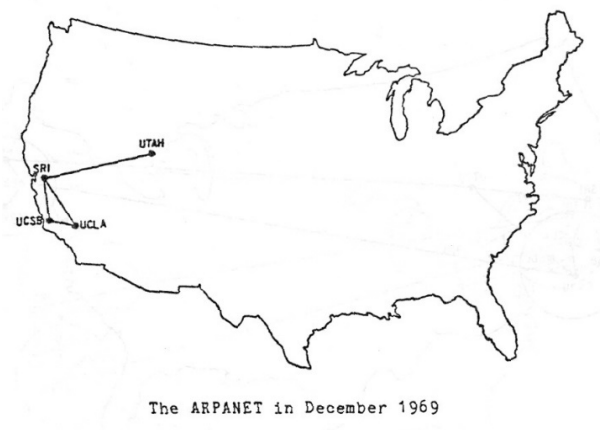
Pada tahun 1964, Paul Baran di Rand Institute mulai melakukan penelitian *packet switching* untuk mendapatkan suara yang aman melalui saluran jaringan militer, dan di Laboratorium Fisika Nasional di Inggris, Donald Davies dan Roger Scantlebury juga mengembangkan gagasan mereka mengenai *packet switching*.

Inovasi dari Kleinrock, Rand, dan NPL telah meletakkan fondasi untuk internet. Tapi internet juga memiliki sejarah panjang dari membangun dan mendemonstrasikan yang kita sebuat *Internet*. JCR Licklider dan Lawrence Roberts, yang merupakan rekan dari Kleinrock's di MIT, melanjutkan untuk memimpin sebuah program di *Advanced Research Projects Agency* (ARPA) di Amerika Serikat. Roberts menerbitkan keseluruhan rencana untuk ARPAnet, jaringan komputer paket pertama yang dilengkapi komputer dan menjadi nenek moyang langsung dari internet publik. Pada Hari Buruh tahun 1969, *packet switching* pertama dipasang di UCLA di bawah pengawasan Kleinrock, dan tiga *packet switching* tambahan dipasang segera di Stanford Stanford Institute (SRI), UC Santa Barbara, dan University of Utah, seperti terlihat pada Gambar 2.1. Pada tahun 1972, ARPAnet telah berkembang menjadi sekitar lima belas *node*. Demonstrasi publik pertama dilakukan oleh Robert Kahn. *Protocol host-to-host* pertama antara sistem akhir ARPAnet yang dikenal sebagai n*etwork-control protocol* (NCP). Dengan *protocol end-to-end*, Ray Tomlinson menulis program *e-mail* pertama di tahun 1972.

2.2.2 Jaringan Internet: 1972-1980

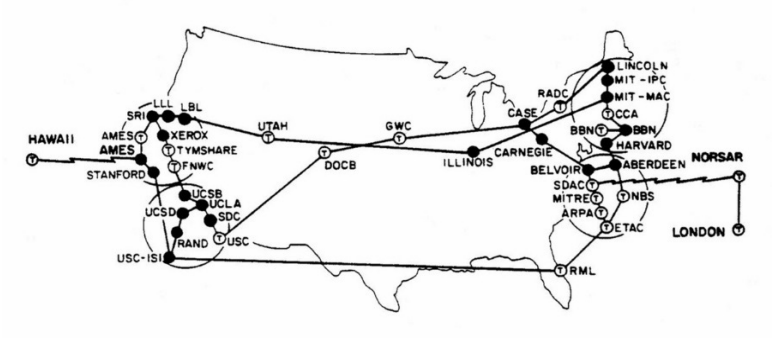
ARPAnet awal adalah jaringan tunggal tertutup. Agar bisa berkomunikasi dengan *host* ARPAnet, seseorang harus benar-benar terhubung dengan IMPA ARPAnet lainnya seperti ditunjukkan pada Gambar 2.2. Pada awal hingga pertengahan 1970-an, jaringan *packet-switching* tambahan yang berdiri sendiri selain ARPAnet adalah

ALOHANet, jaringan *microwave* yang menghubungkan universitas di kepulauan Hawaii, serta paket satelit DARPA dan jaringan packet-radio. Vinton Cerf dan Robert Kahn, di bawah sponsor *Defense Advanced Research Projects Agency* (DARPA), menciptakan jaringan antarjaringan, yang diistilahkan *internetting*.



Gambar 2.2 Arpnet di tahun 1960-an

Versi awal *transmission control protocol* (TCP), sangat berbeda dengan TCP hari ini. Versi awal TCP menggabungkan pengiriman data dalam urutan yang andal melalui transmisi ulang sistem akhir dengan fungsi penerusan yang sekarang dilakukan oleh IP. Percobaan awal dengan TCP, dikombinasikan dengan pengakuan akan pentingnya layanan transportasi *end-to-end* yang tidak dapat diandalkan, tidak mengalir, untuk aplikasi seperti suara *packetized*, menyebabkan pemisahan IP dari TCP dan pengembangan *user datagram protocol* (UDP), tiga protokol internet utama yang kita lihat saat ini TCP, UDP, dan IP secara konseptual ada pada akhir tahun 1970-an.



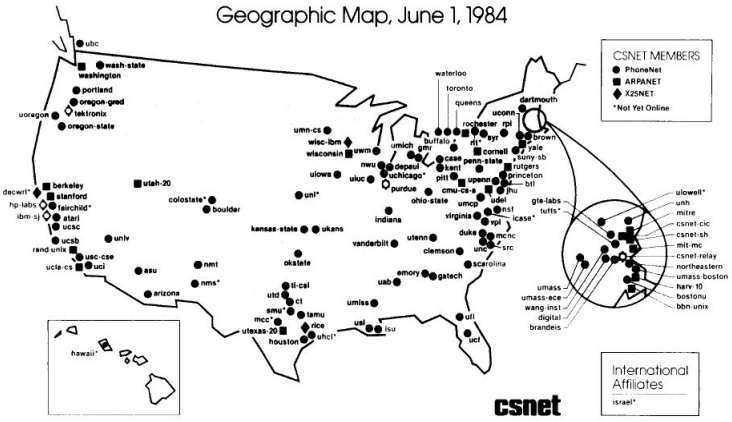
Gambar 2.3 DARPA di Tahun 1960-an

Selain penelitian terkait internet DARPA, seperti terlihat pada Gambar 2.3, di Hawaii, Norman Abramson sedang mengembangkan ALOHAnet, sebuah jaringan radio berbasis paket yang memungkinkan beberapa situs jarak jauh di Kepulauan Hawaii saling berkomunikasi satu sama lain. Protokol ALOHA adalah protokol *multiple-access* pertama, yang memungkinkan pengguna yang didistribusikan secara geografis untuk berbagi media komunikasi tunggal di frekuensi radio. Metcalfe dan Boggs dibangun di atas protokol *multiple-access* saat mereka mengembangkan protokol Ethernet untuk jaringan siaran bersama berbasis kabel. Menariknya, protokol Ethernet Metcalfe dan Boggs dimotivasi oleh kebutuhan untuk menghubungkan beberapa PC, printer, dan *disk* secara bersama. Jauh sebelum revolusi PC dan ledakan jaringan, Metcalfe dan Boggs telah meletakkan dasar bagi PC LAN hingga hari ini.

2.2.3 Jaringan Internet: 1980-1990

Pada akhir tahun 1970-an, sekitar dua ratus *host* terhubung ke ARPAnet dan pada akhir tahun 1980-an jumlah *host* yang terhubung ke internet publik akan mencapai seratus ribu, seperti ditunjukkan pada Gambar 2.4. Tahun 1980-an akan menjadi masa pertumbuhan yang luar

biasa. Sebagian besar pertumbuhan itu dihasilkan dari beberapa upaya berbeda untuk menciptakan jaringan komputer yang menghubungkan universitas secara bersama-sama. BITNET menyediakan *e-mail* dan transfer *file* di antara beberapa universitas di Northeast regional.



Gambar 2.4 ARPANET menjadi internet

CSNET dibentuk untuk menghubungkan peneliti universitas yang tidak memiliki akses ke ARPAnet. Pada tahun 1986, NSFNET diciptakan untuk menyediakan akses ke pusat superkomputer yang disponsori NSF. Dimulai dengan kecepatan *backbone* awal 56 kbps, tulang punggung NSFNET berjalan pada 1,5 Mbps akan berfungsi sebagai tulang punggung utama yang menghubungkan jaringan di komunitas ARPAnet, banyak arsitektur internet mulai berkembang. Pada 1 Januari 1983, penyebaran resmi TCP/IP sebagai protokol *host* standar baru untuk ARPAnet (menggantikan protocol NCP) di mana transisi dari NCP ke TCP semua *host* diharuskan untuk mentransfer alamatnya ke TCP/IP. Pada akhir 1980-an, ekstensi penting dilakukan pada TCP untuk menerapkan kontrol kongesti berbasis host. DNS yang

digunakan untuk memetakan antara nama internet yang dapat dibaca manusia (misalnya, uvers.ac.id).

Pada awal 1980-an, Prancis meluncurkan proyek Minitel, sebuah rencana ambisius untuk membawa jaringan data ke setiap rumah. Disponsori oleh pemerintah Prancis, sistem Minitel terdiri dari jaringan *packet-switched* publik dan server Minitel. Minitel sukses besar pada tahun 1984 ketika pemerintah Prancis menyerahkan sebuah terminal Minitel gratis ke setiap rumah tangga di Prancis bagi yang menginginkannya. Situs Minitel termasuk situs gratis seperti situs direktori telepon dan juga situs pribadi, yang mengumpulkan biaya berbasis penggunaan dari setiap pengguna. Pada puncaknya di pertengahan tahun 1990-an, ia menawarkan lebih dari 20.000 layanan, mulai dari perbankan, rumahan hingga *database* khusus.

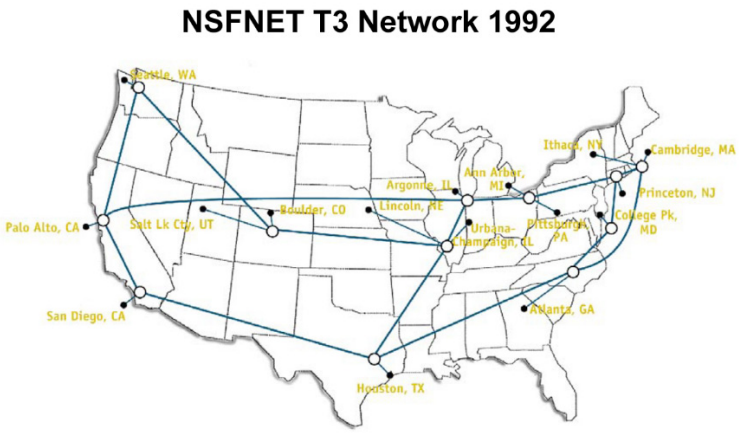
2.2.4 Ledakan Internet: 1990-an

Tahun 1990-an diantar dengan sejumlah peristiwa yang melambangkan evolusi lanjutan dan komersialisasi Internet. ARPAnet, nenek moyang internet, tidak ada lagi. Pada tahun 1991, NSFNET seperti terlihat pada Gambar 2.5, mencabut pembatasan penggunaan NSFNET untuk tujuan komersial. NSFNET sendiri akan dinonaktifkan pada tahun 1995, dengan lalu lintas internet *backbone* dibawa oleh penyedia layanan internet komersial.

Di tahun 1990-an adalah munculnya aplikasi *World Wide Web* (WWW), membawa internet ke rumah dan bisnis jutaan orang di seluruh dunia. Web berfungsi sebagai platform untuk mengaktifkan dan menerapkan ratusan aplikasi baru, termasuk pencarian (misalnya Google dan Bing) perdagangan internet (misalnya Amazon, Lazada, Bukalapak, Tokopedia, dan eBay) dan jejaring sosial (misalnya Facebook).

Web diciptakan di CERN oleh Tim Berners-Lee antara 1989 dan 1991, berdasarkan gagasan yang berasal dari karya sebelumnya

mengenai *hypertext* dari tahun 1940-an oleh Vannevar Bush dan sejak tahun 1960 oleh Ted Nelson.



Gambar 2.5 NSFNET: Tulang Punggung Internet Pertama

Berners-Lee dan rekan-rekannya mengembangkan versi awal HTML, HTTP, server web, dan *browser* yang merupakan empat komponen utama web. Sekitar akhir tahun 1993 ada sekitar dua ratus web server yang beroperasi, koleksi server ini hanya menjadi pertanda dari apa yang akan terjadi selanjutnya. Pada beberapa waktu ini beberapa periset mengembangkan *browser* web dengan antarmuka GUI, termasuk Marc Andreessen, yang bersama Jim Clark, membentuk Mosaic Communications, yang kemudian menjadi Netscape Communications Corporation.

Pada tahun 1995, mahasiswa menggunakan *browser* Netscape untuk menjelajahi web setiap hari. Pada saat ini perusahaan—besar dan kecil—mulai mengoperasikan server web dan melakukan transaksi perdagangan melalui web. Pada tahun 1996, Microsoft mulai membuat *browser*, yang memulai perang *browser* antara Netscape dan Microsoft. Paruh kedua tahun 1990-an adalah periode pertumbuhan dan inovasi

yang luar biasa untuk internet, dengan perusahaan besar dan ribuan pemula menciptakan produk dan layanan internet. Pada akhir milenium, internet mendukung ratusan aplikasi populer, termasuk empat aplikasi pembunuh atau aplikasi distrupsi:

1. *E-mail*, termasuk lampiran dan *e-mail* yang dapat diakses oleh web.
2. Web, termasuk penjelajahan web dan perdagangan internet.
3. Pesan cepat, dengan daftar kontak.
4. *Peer-to-peer file sharing* MP3, dipelopori oleh Napster.

Menariknya, dua aplikasi pembunuh pertama datang dari komunitas riset, sedangkan dua yang terakhir diciptakan oleh beberapa pengusaha muda. Periode dari tahun 1995 sampai 2001 adalah perjalanan *roller coaster* untuk internet di pasar keuangan. Seratusan *startup* internet melakukan penawaran umum perdana dan mulai diperdagangkan di pasar saham. Banyak perusahaan dinilai bernilai miliaran dolar tanpa memiliki arus pendapatan yang signifikan. Saham internet ambruk pada tahun 2000-2001, dan banyak *startups* ditutup. Meski begitu, sejumlah perusahaan muncul sebagai pemenang besar di ruang internet, termasuk Microsoft, Cisco, Yahoo, e-Bay, Google, dan Amazon.

2.2.5 Milenium Baru

Inovasi dalam jaringan komputer terus berlanjut dengan pesat. Kemajuan sedang dilakukan di semua lini, termasuk penyebaran *router* yang lebih cepat dan kecepatan transmisi yang lebih tinggi di jaringan akses dan di tulang punggung jaringan. Tapi perkembangan berikut ini mendapat perhatian khusus:

- Sejak awal milenium, penyebaran akses internet *broadband* agresif ke rumah-rumah, tidak hanya modem kabel dan *digital*

subscriber line (DSL) tapi juga serat optik. Akses internet berkecepatan tinggi ini telah menjadi tulang punggung untuk banyak aplikasi video, termasuk distribusi video buatan pengguna (misalnya YouTube), *streaming* film dan acara televisi sesuai permintaan (misalnya Netflix), dan multi-konferensi video orang (misalnya Skype, Facetime, dan Google Hangouts).

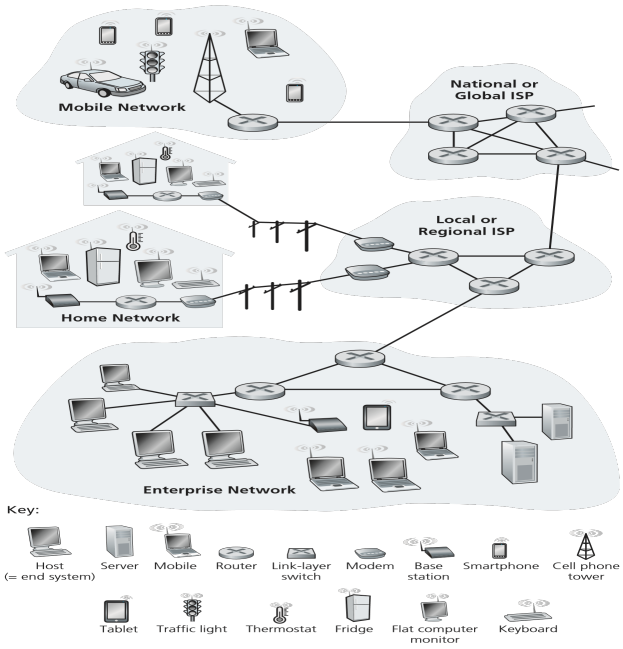
- Meningkatnya kemunculan jaringan WiFi publik berkecepatan tinggi (54 Mbps dan lebih tinggi) dan akses internet berkecepatan menengah (puluhan Mbps) melalui jaringan telepon seluler 4G tidak hanya memungkinkan untuk tetap terhubung terus saat bepergian, namun juga memungkinkan aplikasi spesifik lokasi baru, seperti Maps, Go-jek, Uber, dan Grab. Jumlah perangkat nirkabel yang terhubung ke internet sudah melampaui jumlah perangkat kabel pada tahun 2011.

2.2.6 Apa itu Internet?

Internet hari ini bisa dibilang merupakan sistem rekayasa terbesar dari peradaban manusia, ratusan juta komputer, tautan komunikasi, dan *switch* yang terhubung dengan miliaran pengguna melalui laptop, tablet, dan *smartphone*, seperti terlihat pada Gambar 2.6.

Tapi apa itu internet?

Ada beberapa cara untuk menjawab pertanyaan ini. Pertama, kita bisa menggambarkan internet seperti mur dan baut, yaitu komponen perangkat keras dan perangkat lunak dasar yang membentuk internet itu sendiri. Kedua, kita bisa menggambarkan internet dalam hal infrastruktur jaringan yang memberikan layanan kepada aplikasi terdistribusi.



Gambar 2. 6 Ilustri Kerja Internet (Kurose James F.; Ross, 2013)

Internet adalah jaringan komputer yang menghubungkan miliaran perangkat komputasi di seluruh dunia. Perangkat komputasi ini seperti: PC *desktop*, *workstation*, dan server yang menyimpan dan mengirimkan informasi seperti halaman web dan *e-mail*. Saat ini, semakin banyak perangkat internet nontradisional seperti laptop, *smartphone*, tablet, TV, konsol *game*, termostat, sistem keamanan rumah, peralatan rumah tangga, jam tangan, kacamata, mobil, sistem kontrol lalu lintas dan banyak lagi telah terhubung ke internet. Dalam jargon internet, semua perangkat ini disebut *host* atau sistem akhir. Menurut pencatat aktivitas internet (http://www.internetlivestats.com/) sewaktu buku ini di tulis pada tahun 2018, tercatat hampir sekitar 4 miliar *user*, dan perkiraan jumlahnya akan mencapai 25 miliar perangkat digital di tahun 2020 yang akan terhubung ke internet.

Sistem akhir internet dihubungkan bersama oleh perangkat jaringan yang disebut: *link* komunikasi dan *switch* paket. Ada banyak jenis *link* komunikasi yang terdiri dari berbagai jenis media fisik, termasuk kabel koaksial, kawat tembaga, serat optik, dan spektrum radio. *Link* yang berbeda dapat mengirimkan data pada tingkat yang berbeda, dengan tingkat transmisi *link* diukur dalam *bits/second* (bps). Ketika satu sistem akhir memiliki data untuk dikirim ke sistem akhir yang lain, sistem akhir pengiriman membagi data dan menambahkan *byte header* ke setiap segmen. Paket informasi yang dihasilkan, yang dikenal sebagai paket dalam jargon jaringan komputer, kemudian dikirim melalui jaringan ke sistem akhir tujuan, di mana mereka dirakit kembali menjadi data asli.

2.3 Prinsip-Prinsip Dasar *Internetworking*

Komunikasi di internet melibatkan koordinasi ratusan juta komputer di seluruh dunia. Untuk menghubungkan sistem ini secara bersama-sama, harus memiliki standar untuk segala hal mulai dari pengkabelan fisik sampai dengan protokol aplikasi. Memastikan bahwa sistem mencakup modularitas dan abstraksi sehingga perubahan voltase yang disepakati yang dikirim melalui jaringan tidak mengharuskan insinyur perangkat lunak menulis ulang aplikasi web. Untuk mencapai hal ini, berbagai produsen perangkat lunak dan perangkat keras, badan publik, dan bahkan peretas telah mengembangkan protokol untuk komunikasi. Sejak akhir 1970-an, perancang jaringan telah mengembangkan standar *Layered* (berlapis-lapis), seperti model *open systems interconnection* (OSI, untuk mengkoordinasikan perancangan infrastruktur jaringan.

2.3.1 *Protocol*

Protocol mengambil makna baru jika dilihat dalam konteks penyelidikan forensik. Penyerang berusaha untuk mematahkan dan memutuskan *protocol* untuk menyelundupkan data rahasia, menyelinap

melewati *firewall*, autentikasi *bypass*, dan melakukan serangan seperti *denial-of-service* (DoS). Sementara perancang jaringan dan insinyur melihat *protocol* sebagai aturan untuk memfasilitasi komunikasi antarstasiun. Investigator *network forensics* harus melihatnya sebagai pedoman yang dapat dimanfaatkan penyerang untuk menghasilkan hasil yang tidak diharapkan.

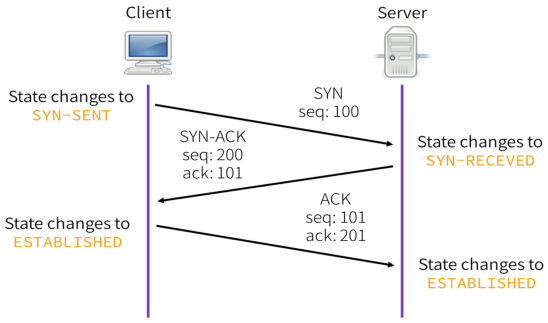
Ilustrasi dari *protocol* adalah seorang Diplomat Jepang dan Amerika dijadwalkan untuk bertemu satu sama lain dan bekerja sama untuk pertama kali. Sayangnya, tidak ada penjelasan singkat tentang kebiasaan sopan santun antarkedua pihak. Pada saat Diplomat Jepang itu menunduk, sangat hormat dan mendalam, di sisi lain pada saat yang sama Diplomat Amerika dengan antusias menyodorkan tangannya. Sayangnya, ketidaktahuan gabungan kedua hasil itu membuat negosiasi tidak berjalan dengan baik.

Setiap komputer yang dirancang untuk berkomunikasi dengan komputer lain melalui jaringan data yang harus bisa mengatasi masalah yang sama seperti ilustrasi di atas. Mereka pertama-tama harus diprogram untuk menggunakan beberapa skema terkoordinasi untuk pertukaran data, dari "halo" menjadi "selamat tinggal" untuk *protocol* komunikasi. Selanjutnya, mereka harus memiliki sistem untuk menangani pesan ambigu dan kondisi yang keliru karena tidak mungkin menentukan reaksi untuk setiap tindakan yang mungkin terjadi.

Menurut *The Free On-line Dictionary of Computing* mendefinisikan *protocol* sebagai: satu set aturan formal yang menjelaskan bagaimana mentransmisikan data, terutama di jaringan. Beberapa *protocol* cukup sederhana. Sebagai contoh, bayangkan bahwa saya menghubungi nomor Anda dan telepon Anda berdering, menandakan kepada Anda bahwa saya ingin berkomunikasi. Anda mengakui permintaan saya dan komunikasi memfasilitasi dengan saya dengan mengambil gagang telepon dan berkata, "Halo, Pak Imam Riadi!" Saya mengakui sinyal

dengan mengatakan sesuatu, seperti "Hi, Mas Ade ..." Dan dalam tiga langkah mudah kita sudah melakukan percakapan dua arah.

Transmission control protocol (TCP) menggunakan tiga langkah yang berbeda untuk membentuk komunikasi dua arah yang andal antar-*host*. Komputer pengirim mengirimkan *set flag* TCP ke komputer penerima dengan sebuah segmen SYN ("sinkronisasi"), menandakan bahwa antarkomputer/*host* ingin berkomunikasi. Komputer penerima mengakui permintaan komputer pengirim dan menfasilitasi komunikasi dengan mengirimkan segmen TCP dengan segmen SYN/ACK ("menyinkronkan"/"*acknowledgment*"). Komputer pengirim mengenali sinyal ini dengan merespons segmen yang memiliki flag ACK TCP ("*acknowledgement*"). Dalam tiga langkah, antara dua komputer telah melakukan jabat tangan TCP dan mendirikan saluran komunikasi yang andal, seperti terlihat pada Gambar 2.8 di bawah ini.



Gambar 2.7 *Transmission Control Protocol Three Handshake*

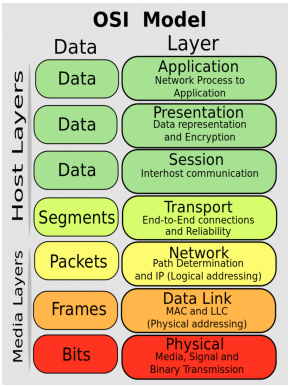
Internet Engineering Task Force (IETF) dan badan-badan standar lainnya telah menerbitkan ribuan standar *protocol* untuk merumuskan poin-poin penting dari *protocol* itu seperti di bawah ini:

- aturan untuk membuat komunikasi menjadi sukses antar sistem yang berbeda,

- spesifikasi yang sudah diatur sebelumnya,
- dirancang untuk menghindari ketergantungan implementasi,
- dirancang untuk menghindari ambiguitas,
- dirancang untuk pulih dengan cepat dari kesalahan.

2.3.2 Model *Open Systems Interconnection* (OSI)

Model *open systems interconnection* (OSI) seperti terlihat pada Gambar 2.9 dirancang oleh *International Organization for Standardization* (ISO) untuk menyediakan arsitek, insinyur perangkat lunak, dan produsen perangkat keras dan lunak dengan kerangka modular yang fleksibel untuk pengembangan sistem komunikasi.



Gambar 2.8 Model *Open Systems Interconnection* (OSI) *Layer*

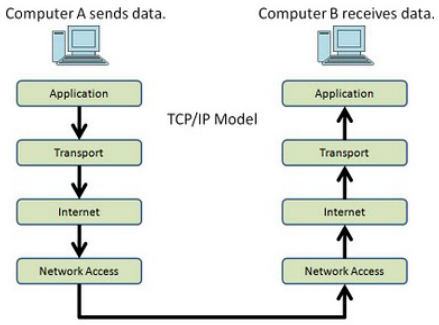
Ketika data dikirim ke jaringan, *output* dari satu lapisan *dienkapsulasi* dengan data yang dirancang untuk digunakan oleh proses lapisan bawah. Sebaliknya, ketika data diterima oleh *host* tujuan, masukan dari setiap lapisan *demultiplexed* untuk digunakan oleh proses lapisan yang lebih tinggi. Hal ini mirip dengan cara orang menulis surat, huruf-huruf dipecah-pecah (*dienkapsulasi*) lalu dimasukan ke dalam amplop dan menulis alamat penerima di bagian luar, dengan huruf-huruf yang telah

dipecah terus di komputer penerima dilakukan *demultiplexed* untuk merangkai isi dari surat tersebut. Demikian pula komputer, meng-*encapsulate* (merangkum atau membungkus) data dari satu lapisan dengan *header* standar dan/atau *footer* sehingga perangkat lunak pada lapisan bawah dapat mengolahnya. Ini adalah bagaimana sebuah prose data dikirimkan dari satu server ke server lain di Internet. Bila data diterima oleh server tujuan maka *demultiplexed* atau dibukanya. Proses penerimaan memeriksa metadata *protokasi* untuk menentukan proses lapisan tinggi mana yang akan digunakan, dan kemudian menghapus metadata *protocol* lapisan saat ini sebelum mengirimnya.

Tantangan yang sangat kompleks sering kali dapat dipecahkan dengan lebih mudah jika dipecah menjadi beberapa bagian. Dengan menggunakan model OSI, atau kerangka kerja serupa, seperti Model TCP/IP, menggunakan empat lapisan bukan tujuh. Manfaat besar dari pendekatan berlapis adalah walaupun mungkin ada beberapa *protocol* bersaing yang digunakan untuk satu tujuan, ada lapisan abstraksi dan modularitas yang memungkinkan kita menukar solusi pada satu lapisan tanpa memengaruhi salah satu di lapisan lainnya.

2.3.3 *Internet ProtocolSuite* (TCP/IP)

Internet protocolsuite seperti telihat pada Gambar 2.10 atau dikenal juga sebagai protocol TCP/IP adalah kumpulan *protocol* yang digunakan untuk mengimplementasikan fungsi penting di internet dan di banyak jaringan *packet-switched* lainnya. Untuk investigator *network forensics*, *protocol* yang membentuk *internet protocolsuite* (TCP/IP) sangat penting. Efektivitas sebagai penyelidik *network forensics* akan sebagian bergantung pada keakraban dengan *internet protocolsuite* ini, termasuk: *header,* analisis *flow record* sampai analisis paket hingga *proxy* web, dan banyak lagi.



Gambar 2.9 TCP/IP *Protocol*

(Vacca, 2009)

2.3.4 *Internet Protocol* (IP)

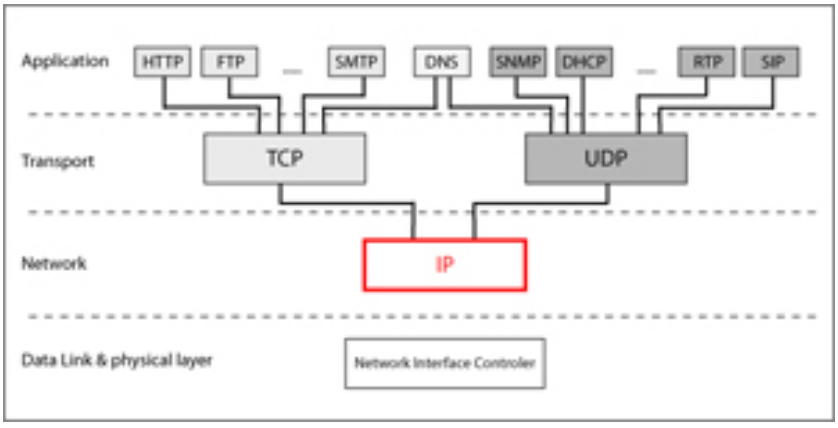
Internet protocol (IP) dirancang untuk menangani pengalamatan dan perutean. Mencakup metode identifikasi untuk sumber dan *protocol* tujuan unik pada jaringan dan memberikan dukungan untuk *routing* data melalui jaringan. Untuk mencapai hal ini, data diawali dengan *header* IP. Kombinasi dari *header* IP dan muatan yang *dienkapsulasi* bersama disebut sebagai paket IP. Tidak ada *footer*; panjang paket IP ditentukan di *header*-nya. Paket dikirim dari sumber melalui *router* perantara ke tempat tujuannya, di mana *header* IP dikeluarkan oleh sistem operasi penerima dengan muatan yang *dienkapsulasi*. IP beroperasi pada *layer* 3 dari model OSI (*network layer*), seperti terlihat pada Gambar 2.10 Ini adalah bentuk *protocol* tanpa koneksi, yang berarti bahwa ia tidak secara eksplisit mengidentifikasi paket individual sebagai bagian dari seri terkait, dan karena itu juga tidak memiliki metode untuk menunjukkan inisiasi atau mengakhiri sebuah komunikasi.

IP juga tidak dirancang untuk memastikan keandalan transmisi, para perancang membayangkan bahwa fungsi ini akan ditangani oleh protokol TCP atau lapisan atas yang lebih tinggi. *Internet control message protocol* (ICMP), yang dirilis pada tahun 1981 bersamaan dengan

protokol internet versi 4 (IPv4), sering digunakan bersamaan dengan IP untuk memberi umpan balik jika ditemukan masalah saat perangkat berkomunikasi.

Ada beberapa versi IP. Versi yang paling banyak digunakan adalah IPv4, yang secara resmi distandardisasi pada tahun 1981 dan pertama kali diluncurkan pada tanggal 1 Januari 1983. Selama tiga puluh tahun terakhir, IPv4 telah menjadi pendukung global. Namun, pekerjaan terus pada pengembangan IP, terutama didorong oleh kekhawatiran tentang habisnya slot alamat IP. Pada tahun 1998, spesifikasi untuk *internet protocol*versi 6 (IPv6) dirilis. Karakteristik *internet protocol* (IP) meliputi:

- dukungan untuk pengalamatan dan perutean,
- bisa diandalkan,
- termasuk *header* (tanpa *footer*),
- *header* plus *payload* disebut paket IP.

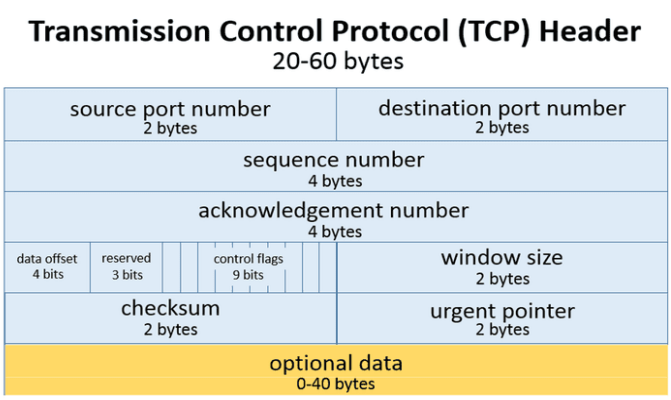


Gambar 2.10 *Internet Protocol* (IP)

2.3.5 *Transmission Control Protocol*

Transmission control protocol (TCP) dirancang untuk menangani komunikasi *multiplexing* proses pada *host*, urutan dan serta keandalan

jaringan. Data diawali dengan *header* TCP, seperti yang ditunjukkan pada Gambar 2.11.



Gambar 2.11 *Transmission Control Protocol Header*

Kombinasi dari *header* TCP dan muatan yang *dienkapsulasi* bersama disebut sebagai segmen TCP. Seperti IP, tidak ada *footer*. Untuk berkomunikasi, sebuah proses *mengenkapsulasi* data di *header* segmen TCP untuk transmisi ke proses lain, dalam hal ini segmen kemudian akan *dienkapsulasi* dalam paket IP untuk transmisi di seluruh jaringan dan *demultiplex* pada *node* ujung di sisi penerima.

Header TCP mencakup bidang 16-byte untuk *source port*/*port* sumber dan *destination port*/*port* tujuan. Sesuai dengan *specification of internet transmission control program*, sebuah proses mungkin memiliki sejumlah *port* yang digunakan untuk berkomunikasi dengan *port* dari proses lainnya dan nilai untuk *port* TCP berkisar antara 0 sampai 65,535. TCP adalah protokol *connection-oriented*, artinya menunjukkan keadaan transmisi dengan menggunakan *flag* pada *header* TCP. Seperti dijelaskan sebelumnya, TCP menggunakan tiga langkah untuk membentuk komunikasi dua arah yang andal antarstasiun. Pertama, stasiun inisiat mengirimkan segmen dengan *flag* SYN. Selanjutnya,

responden mengirim kembali segmen dengan *flag* SYN dan ACK. Akhirnya, *initiator* merespons dengan segmen yang memiliki *flag* ACK. Urutan inisiasi ini disebut sebagai *three-way handshake.* Ada juga urutan dua langkah yang sesuai untuk menutup koneksi menggunakan *flag* FIN dan ACK. Karakteristik utama dari protokol TCP meliputi:

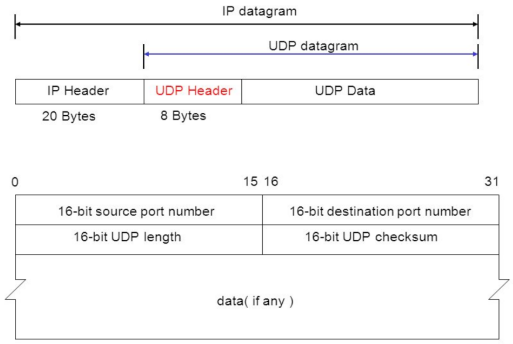
- dapat diandalkan,
- berorientasi koneksi,
- menangani *sekuensing*,
- nomor *port* berkisar antara 0 sampai 65.535,
- termasuk *header* (tanpa *footer*),
- *header plus payload* disebut segmen TCP.

2.3.6 *User Datagram Protocol* (UDP)

User datagram protocol (UDP) adalah sebuah protokol sederhana. Seperti terlihat pada Gambar 2.13, tidak hanya dirancang untuk memudahkan komunikasi *muliplexing* proses pada *host*, keandalan, urutan, pengaturan koneksi dari TCP. Ini sangat berguna untuk beberapa aplikasi, seperti *streaming* data *realtime* (voice over IP, musik, atau video). Untuk aplikasi ini, tidak ada gunanya mencoba mendeteksi jika ada paket tidak berfungsi atau dibatalkan karena persyaratan waktu tidak memberi ruang untuk pengiriman ulang. Seperti TCP, *header* UDP mencakup bidang 16-byte *source port*/*port* sumber dan *destination port*/*port* tujuan. Nilai untuk *port* UDP berkisar antara 0 sampai 65,535.

Karakteristik utama dari protokol UDP meliputi:

- tidak bisa diandalkan,
- nomor *port* berkisar antara 0 sampai 65535,
- termasuk *header* (tanpa *footer*),
- *header plus payload* disebut datagram UDP.



Gambar 2.12 *User Datagram Protocol* (UDP)

(J. M. Kizza, 2013)

2.4 Soal Latihan

1. Jelaskan definisi, sejarah singkat pengembangan dan cara kerja dari *packet switching*!
2. Uraikan secara singkat latar belakang jaringan internet di era: 1972-1980; 1980-1990; ledakan internet: 1990-an; dan cara kerja dari internet!
3. Jelaskan secara singkat cara kerja dari *protocol*!
4. Jelaskan secara singkat cara kerja dari model *open systems interconnection* (OSI)!
5. Jelaskan secara singkat cara kerja dari *internet protocolsuite* (TCP/IP)!
6. Jelaskan secara singkat cara kerja dari *internet protocol* (IP)!
7. Jelaskan secara singkat cara kerja dari *transmission control protocol* (TCP)!

Bab 3
Computer Security

Tujuan Instruksional Umum:

1. Mahasiswa/wi mampu menjelaskan tentang *computer security*.

Tujuan Instruksional Khusus:

1. Mahasiswa/wi mampu memahami tentang *internet security*.
2. Mahasiswa/wi mampu memahami tentang *threat*.
3. Mahasiswa/wi mampu memahami tentang *vulnerability*.
4. Mahasiswa/wi mampu memahami tentang *network security*.

3.1 *Internet Security*

Internet security adalah cabang dari *computer security* yang secara khusus terkait dengan internet yang melibatkan keamanan jaringan pada tingkat yang lebih umum yang berlaku juga untuk aplikasi lain atau sistem operasi secara keseluruhan. Tujuannya adalah untuk menetapkan peraturan dan tindakan untuk melawan serangan terhadap internet. Internet merupakan saluran komunikasi yang tidak aman untuk bertukar informasi yang mengarah pada risiko intrusi atau penipuan

yang tinggi, seperti *phishing*. Metode yang berbeda telah digunakan untuk melindungi transfer data, yaitu salah satunya enkripsi.

Untuk melindungi protokol TCP/IP dapat menggunakan metode kriptografi dan protokol keamanan. Protokol ini termasuk *secure sockets layer* (SSL), *transport layer security* (TLS) untuk lalu lintas web, *pretty good privacy* (PGP) untuk *email*, dan IPsec untuk keamanan lapisan jaringan. *Internet protocol security* (IPsec) dirancang untuk melindungi komunikasi TCP/IP secara aman. *Internet protocol security* (IPsec) adalah seperangkat ekstensi keamanan yang dikembangkan oleh *Internet Task Force* (IETF). IPsec menyediakan keamanan dan autentikasi pada lapisan IP dengan mengubah data menggunakan enkripsi. Dua jenis transformasi utama yang menjadi dasar IPsec: *authentication header* (AH) dan *encapsulating security payload* (ESP).

Kedua protokol ini menyediakan integritas data, autentikasi asal data, dan layanan anti-*replay*. Protokol ini dapat digunakan sendiri atau dikombinasikan untuk menyediakan seperangkat layanan keamanan yang diinginkan untuk lapisan protokol internet (IP).

Komponen dasar arsitektur keamanan IPsec dijelaskan dalam bentuk fungsi berikut:

- protokol keamanan untuk AH dan ESP,
- asosiasi keamanan untuk pengelolaan kebijakan dan pemrosesan lalu lintas,
- manajemen kunci manual dan otomatis untuk *internet key exchange* (IKE),
- *algoritma* untuk autentikasi dan enkripsi.

Kumpulan layanan keamanan yang disediakan di lapisan IP mencakup kontrol akses, integritas asal data, perlindungan terhadap tayangan ulang, dan kerahasiaan. *Algoritma* ini memungkinkan perangkat ini bekerja secara independen tanpa memengaruhi bagian

lain dari implementasi. Implementasi IPsec dioperasikan di lingkungan *host* atau *gateway* keamanan yang memberikan perlindungan terhadap lalu lintas IP.

Beberapa situs *online* menawarkan kepada pelanggan kemampuan untuk menggunakan kode digit yang secara acak berubah setiap 30-60 detik pada *security token* atau token keamanan. Kunci pada token keamanan telah dibangun dalam perhitungan matematis dan memanipulasi angka berdasarkan waktu saat ini yang terpasang pada perangkat. Ini berarti bahwa setiap tiga puluh detik hanya ada sejumlah angka tertentu yang mungkin benar untuk memvalidasi akses ke akun *online*.

Gambar 3.1 *Security Token*

Setiap pengguna situs web yang masuk seperti yang terlihat pada Gambar 3.1, akan dikenali dari nomor seri perangkat dan menghitung waktu untuk memverifikasi bahwa token yang diberikan memang merupakan pengguna yang sah. Setelah 30-60 detik perangkat akan menampilkan nomor enam digit acak baru yang bisa masuk ke situs web. Metode pengamanan komunikasi dalam jaringan digunakan untuk melindungi transfer data, termasuk enkripsi yang beroperasi di bagian

lapisan paling atas dari model OSI. Paket jaringan diteruskan hanya jika sambungan dibuat menggunakan protokol yang dikenal. *Gateway* akan menganalisis keseluruhan paket data individual saat data dikirim atau diterima.

Pesan *email* disusun, disampaikan, dan disimpan dalam beberapa proses langkah, yang dimulai dengan komposisi pesan. Saat pengguna menyelesaikan penulisan pesan dan mengirimkannya, pesan tersebut diubah menjadi format standar: pesan berformat RFC 2822. Setelah itu, pesan bisa ditransmisikan. Dengan menggunakan koneksi jaringan, klien *email*, disebut *mail user agent* (MUA), terhubung ke *mail transfer agent* (MTA) yang beroperasi di server surat. Klien *email* kemudian memberikan identitas pengirim ke server. Selanjutnya, dengan menggunakan perintah server surat, klien mengirimkan daftar penerima ke server surat. Klien kemudian memasok pesannya. Begitu server surat menerima dan memproses pesan, beberapa kejadian terjadi: identifikasi server penerima, pembentukan koneksi, dan pengiriman pesan. Dengan menggunakan layanan *Domain Name System* (DNS), server surat pengirim menentukan server surat untuk penerimanya. Kemudian, server membuka koneksi ke server surat penerima dan mengirim pesan menggunakan sebuah proses yang serupa dengan yang digunakan oleh klien asal, mengirimkan pesan ke penerima.

Pretty good privacy (PGP) dapat memberikan kerahasiaan dengan mengenkripsi pesan yang dikirim atau *file* data tersimpan dengan menggunakan *algoritma* enkripsi seperti Triple DES atau CAST-128. Pesan *email* dapat dilindungi dengan menggunakan kriptografi dengan berbagai cara, seperti berikut ini:

- Menandatangani pesan *email* untuk memastikan integritasnya dan mengonfirmasi identitas pengirimnya.
- Mengenkripsi isi pesan *email* untuk memastikan kerahasiaannya.

- Mengenkripsi komunikasi antara server *email* untuk melindungi kerahasiaan dari kedua pesan dan *header* pesan.

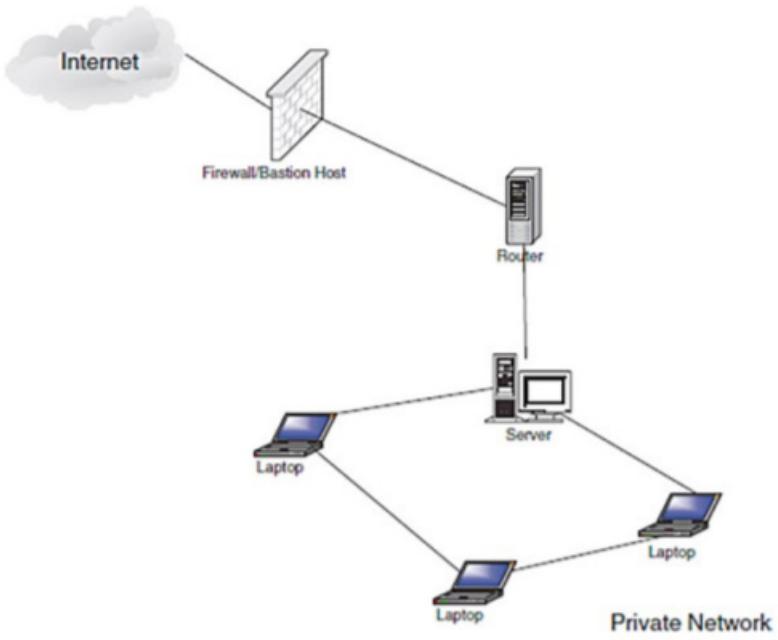
Dua metode pertama, penyandian pesan dan penyandian pesan, sering digunakan bersamaan. Namun, mengenkripsi transmisi antarserver *email* biasanya digunakan hanya bila dua organisasi ingin melindungi *email* yang secara teratur dikirim satu sama lain. Misalnya, organisasi dapat membuat *virtual private network* (VPN) untuk mengenkripsi komunikasi antarserver *email* mereka melalui internet. Tidak seperti metode yang hanya dapat mengenkripsi badan pesan, VPN dapat mengenkripsi seluruh pesan, termasuk informasi *header email*, seperti pengirim, penerima, dan subjek. Dalam beberapa kasus, organisasi mungkin perlu melindungi informasi *header*. Namun, solusi VPN saja tidak dapat menyediakan mekanisme penandatanganan pesan, juga tidak dapat memberikan perlindungan untuk pesan *email* sepanjang rute dari pengirim ke penerima.

Multipurpose internet mail extensions (MIME) mentransformasikan data non-ASCII di situs pengirim ke data *network virtual terminal* (NVT) ASCII dan mengirimkannya ke *simple mail transfer protocol* (SMTP) untuk dikirim melalui internet. SMTP server di sisi penerima menerima data NVT ASCII dan mengirimkannya ke MIME untuk diubah kembali ke data non-ASCII asli.

Message authentication code (MAC) adalah metode kriptografi yang menggunakan kunci rahasia untuk mengenkripsi pesan. Metode ini menghasilkan nilai MAC yang dapat didekripsi oleh *receiver*, menggunakan kunci rahasia yang sama yang digunakan oleh pengirim. Kode autentikasi pesan melindungi integritas data pesan sekaligus keasliannya.

Firewall komputer mengendalikan akses antarjaringan. Umumnya terdiri dari *gateway* dan *filter* yang bervariasi dari satu *firewall* ke *firewall* lainnya. *Firewall* juga memonitor lalu lintas jaringan dan mampu

memblokir lalu lintas yang berbahaya. *Firewall* bertindak sebagai server perantara antara koneksi SMTP dan *Hypertext Transfer Protocol* (HTTP).



Gambar 3.2 *Firewall*

(J. M. Kizza, 2013)

Peran *firewall* dalam keamanan web seperti terlihat pada Gambar 3.2 adalah pemberlakukan batasan paket jaringan masuk dan keluar ke dan dari jaringan pribadi. Lalu lintas masuk atau keluar harus melewati *firewall*. Hanya lalu lintas resmi yang diizinkan melewatinya. *Firewall* membuat pos pemeriksaan antara jaringan pribadi internal dan internet publik. *Firewall* dapat membuat suatu aturan berdasarkan sumber IP dan nomor *port* TCP. Mereka juga bisa berfungsi sebagai platform untuk IPsec. Dengan menggunakan kemampuan *tunneling*, *firewall*

bisa digunakan untuk mengimplementasikan VPN. *Firewall* juga bisa membatasi eksposur jaringan dengan menyembunyikan sistem jaringan internal dan informasi dari internet publik. Terdapat tiga jenis *firewall*:

1. *Packet Filter*

   Firewall generasi pertama yang memproses lalu lintas jaringan berdasarkan paket per paket. Tugas utamanya adalah untuk menyaring lalu lintas dari *host IP remote*, jadi *router* diperlukan untuk menghubungkan jaringan internal ke internet. *Router* dikenal sebagai *router screening*, yang membuat paket layar meninggalkan dan memasuki jaringan.

2. *Stateful Packet Inspection*

   Server *proxy* yang beroperasi pada tingkat jaringan model *open system interconnection* (OSI) dan secara statis menentukan lalu lintas yang akan diizinkan. *Proxy* sirkuit akan meneruskan paket jaringan yang berisi nomor *port* yang diberikan. Keuntungan utama dari server *proxy* adalah kemampuannya untuk menyediakan *network address translation* (NAT), yang dapat menyembunyikan alamat IP pengguna dari internet, yang secara efektif melindungi semua informasi internal dari internet.

3. *Application-level Gateway*

   Firewall generasi ketiga di mana server *proxy* beroperasi di bagian paling atas model OSI, tingkat TCP/IP. Paket jaringan hanya diteruskan jika sambungan dibuat menggunakan protokol yang dikenal. *Gateway* tingkat aplikasi terkenal untuk menganalisis keseluruhan pesan daripada paket data individual saat data dikirim atau diterima.

Seorang pengguna komputer dapat tertipu atau dipaksa untuk men-*download* perangkat lunak ke komputer yang memiliki maksud jahat. Perangkat lunak semacam itu hadir dalam berbagai bentuk,

seperti *virus, trojan horse, spyware,* dan *worm*. *Malware* kependekan dari *malicious software* atau perangkat lunak berbahaya. *Malware* adalah perangkat lunak yang digunakan untuk mengganggu operasi komputer, mengumpulkan informasi sensitif, atau mendapatkan akses ke sistem komputer pribadi. *Malware* didefinisikan oleh maksud jahatnya, yang bertentangan dengan persyaratan pengguna komputer yang menyertakan perangkat lunak menyebabkan kerusakan. Istilah *badware* kadang-kadang digunakan, dan diterapkan pada *malware* yang benar-benar jahat.

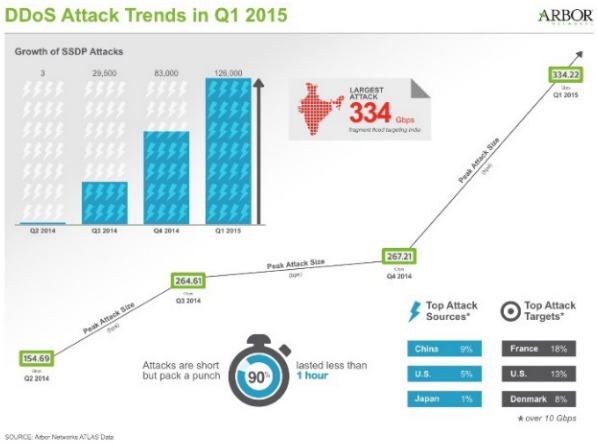
Berikut ini adalah tipe-tipe dari *malware*:

1. Botnet adalah jaringan komputer zombie yang telah diambil alih oleh robot atau bot yang melakukan tindakan berbahaya berskala besar untuk pencipta botnet.
2. Virus komputer adalah program yang dapat mereplikasi struktur atau efeknya dengan menginfeksi *file* atau struktur lain di komputer. Penggunaan virus secara umum adalah mengambil alih komputer untuk mencuri data.
3. *Worm* komputer adalah program yang dapat mereplikasi dirinya di seluruh jaringan komputer, melakukan tugas berbahaya di seluruh dunia.
4. *Ransomware* adalah jenis *malware* yang membatasi akses ke sistem komputer yang terinfeksi, dan menuntut uang tebusan yang dibayarkan kepada pencipta *malware* agar pembatasan dihapus.
5. *Scareware* adalah perangkat lunak *scam* dengan muatan berbahaya, biasanya terbatas atau tidak ada manfaatnya, yang dijual ke konsumen melalui praktik pemasaran tidak etis tertentu. Pendekatan penjualan menggunakan *social engineering*/ rekayasa sosial untuk menimbulkan kejutan, kegelisahan, atau

persepsi ancaman, yang umumnya ditujukan pada pengguna yang tidak menaruh curiga.

6. *Spyware* mengacu pada program yang diam-diam memantau aktivitas pada sistem komputer dan melaporkan informasi tersebut kepada orang lain tanpa sepengetahuan pengguna.
7. *Trojan Horse*, biasa dikenal dengan *Trojan* adalah istilah umum untuk perangkat lunak berbahaya yang berpura-pura tidak berbahaya, sehingga pengguna mengizinkannya untuk diunduh ke komputer.

Serangan *denial-of-service* (DoS) atau serangan *distributed denial-of-service* (DDoS) adalah upaya untuk membuat sumber daya komputer tidak tersedia bagi pengguna yang sah. Meskipun sarana untuk melakukan, motif, dan target serangan DoS dapat bervariasi, namun umumnya terdiri dari upaya bersama untuk mencegah situs atau layanan internet berfungsi dengan baik atau tidak untuk sementara atau tanpa batas waktu.



Gambar 3.3 Laporan DDoS 2015 (Symantec, 2017)

Menurut pelaku bisnis yang berpartisipasi dalam survei keamanan bisnis internasional, 25% responden mengalami serangan mengalami peningkatan per tahunnya seperti terlihat pada Gambar 3.3.

Phishing adalah tempat penyerang berpura-pura menjadi entitas yang dapat dipercaya, baik melalui *email* atau halaman web. Korban diarahkan ke halaman web palsu agar terlihat sah, melalui *email* palsu, *instant messenger*/media sosial atau jalan lainnya. Sering kali taktik seperti *email spoofing* digunakan untuk membuat *email* tampak berasal dari pengirim yang sah, atau subdomain kompleks yang panjang menyembunyikan *host* situs sebenarnya. Kelompok asuransi RSA mengatakan bahwa *phishing* menyumbang kerugian di seluruh dunia sebesar $1,5 Miliar pada tahun 2012.

Statistik *browser* web cenderung memengaruhi jumlah *browser* web yang dieksploitasi. Sebagai contoh Internet Explorer 6 yang digunakan mayoritas pangsa pasar *browser* web yang dianggap sangat tidak aman karena kerentanan. Kerentanan dieksploitasi di banyak browser yang berbeda, terlihat Internet Explorer di 28,5%, Firefox di 18,4%, Google Chrome sebesar 40,8%.

3.2 *Threat*

Kejahatan komputer atau *cybercrime* adalah kejahatan yang melibatkan komputer dan jaringan. Komputer mungkin telah digunakan dalam pelaksanaan kejahatan, atau mungkin menjadi sasaran serangan. Kejahatan semacam itu dapat mengancam stabilitas sosial politik, keamanan, kesehatan dan finansial suatu negara. Jenis kejahatan ini telah menjadi profil tinggi, terutama *hacking*, pelanggaran hak cipta, dan pornografi anak. Ada juga masalah privasi, saat informasi pribadi yang bersifat rahasia dicegat atau diungkapkan secara tidak sah dengan motif membahayakan korban secara psikologis dan fisik, menggunakan

jaringan telekomunikasi modern seperti internet dan telepon genggam.

Secara internasional, baik aktor pemerintah maupun non-negara terlibat dalam *cybercrimes*, termasuk spionase, pencurian keuangan, dan kejahatan lintas batas lainnya. Kegiatan yang melintasi batas internasional dan melibatkan kepentingan setidaknya satu negara bangsa terkadang disebut sebagai *cyberwarfare*. Sistem hukum internasional berusaha meminta pertanggungjawaban aktor tersebut atas tindakan mereka melalui Pengadilan Pidana Internasional. Sebuah laporan dari McAfee memperkirakan bahwa kerusakan tahunan pada ekonomi global mencapai 445 miliar USD. Namun, sebuah laporan Microsoft menunjukkan bahwa perkiraan berbasis survei semacam itu telah membesar-besarkan kerugian yang sebenarnya.

Computer fraud adalah salah representasi fakta yang tidak jujur yang dimaksudkan untuk membiarkan orang lain melakukan sesuatu yang menyebabkan kerugian. Dalam konteks ini, kecurangan akan menghasilkan keuntungan dengan:

- Mengubah dengan cara yang tidak sah. Ini memerlukan sedikit keahlian teknis dan merupakan bentuk pencurian umum oleh karyawan yang mengubah data sebelum masuk atau memasukkan data palsu, atau dengan memasukkan instruksi yang tidak sah atau menggunakan proses yang tidak sah;
- Mengubah, menghancurkan, menekan, atau mencuri, biasanya untuk menyembunyikan transaksi yang tidak sah;
- Mengubah atau menghapus data yang tersimpan.

Bentuk kecurangan lainnya dapat difasilitasi dengan menggunakan sistem komputer, termasuk penipuan bank, *carding*, pencurian identitas, pemerasan, dan pencurian informasi. Berbagai penipuan internet, banyak berbasis *phishing* dan *social engineering*, target konsumen dan bisnis.

Pejabat pemerintah dan spesialis keamanan teknologi informasi telah mendokumentasikan peningkatan yang signifikan dalam masalah internet dan pemindaian server sejak awal 2001. Namun, ada kekhawatiran yang berkembang di berbagai kalangan bahwa penyusupan semacam itu merupakan bagian dari upaya terorganisir oleh pelaku *cyberterror*, dinas intelijen asing, atau kelompok lainnya untuk memetakan celah keamanan potensial. Seorang *cyberterrorist* adalah seseorang yang mengintimidasi atau menggalang pemerintah atau organisasi untuk memajukan tujuan politik atau sosialnya dengan meluncurkan serangan berbasis komputer terhadap komputer, jaringan, atau informasi yang tersimpan di dalamnya.

Cyberterrorisme secara umum dapat didefinisikan sebagai tindakan terorisme yang dilakukan melalui penggunaan dunia maya atau sumber daya komputer. Dengan begitu, sebuah propaganda sederhana di internet bahwa akan terjadi serangan bom selama liburan bisa dianggap *cyberterrorism*. Selain itu juga ada aktivitas *hacking* yang ditujukan pada individu, keluarga, yang diselenggarakan oleh kelompok-kelompok di dalam jaringan, cenderung menimbulkan ketakutan di kalangan orang-orang, menunjukkan kekuatan, mengumpulkan informasi yang relevan untuk menghancurkan kehidupan masyarakat, perampokan, dan pemerasan.

Cyberextortion terjadi saat sebuah situs web, server *e-mail*, atau sistem komputer dikenai atau diancam dengan penolakan berulang terhadap layanan atau serangan lainnya oleh *hacker*. Peretas ini menuntut uang sebagai imbalan untuk menghentikan serangan dan menawarkan "perlindungan". Menurut FBI, para pelaku *cybernetologist* semakin menyerang situs web perusahaan dan jaringan, melumpuhkan kemampuan mereka untuk mengoperasikan dan menuntut pembayaran untuk memulihkan layanan mereka. Lebih dari 20 kasus dilaporkan setiap bulan ke FBI dan banyak yang tidak dilaporkan untuk menjaga

agar nama baik korban tetap terjaga dari domain publik. Pelaku biasanya menggunakan serangan DDoS. Contoh *cyberextortion* adalah serangan terhadap Sony Pictures tahun 2014.

Department of Defense (DoD) Amerika Serikat, mencatat bahwa dunia maya telah muncul sebagai perhatian tingkat nasional melalui beberapa peristiwa terkini mengenai signifikansi geo-strategis. Di antaranya serangan terhadap infrastruktur Estonia di tahun 2007, yang diduga oleh *hacker* Rusia. Pada bulan Agustus 2008, Rusia kembali melakukan serangan *cyber*, kali ini dalam kampanye kinetik dan non-kinetik yang terkoordinasi dan disinkronkan melawan negara Georgia. Khawatir bahwa serangan semacam itu bisa menjadi perang di masa depan di antara negara-bangsa.

Kejahatan komputer sebagai target dilakukan oleh kelompok kriminal terpilih. Tidak seperti kejahatan yang menggunakan komputer sebagai alat, kejahatan ini memerlukan pengetahuan teknis pelaku. Kejahatan ini relatif baru, yang selama ini hanya ada selama komputer—yang menjelaskan bagaimana masyarakat yang tidak siap dan dunia pada umumnya memerangi kejahatan ini. Ada banyak kejahatan komputer sebagai target ini yang dilakukan setiap hari di internet. Kejahatan yang terutama menargetkan jaringan komputer atau perangkat meliputi:

- Virus komputer
- *Denial-of-service attacks*
- *Malware (malicious code)*

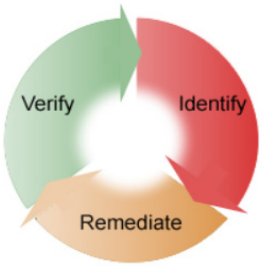
Bila individu merupakan target utama *cybercrime* (kejahatan komputer sebagai target), komputer bisa dianggap sebagai alat ketimbang target. Kejahatan ini umumnya kurang melibatkan keahlian teknis. Kelemahan manusia umumnya dieksploitasi. Kerusakan yang ditangani sebagian besar bersifat psikologis dan tidak berwujud, membuat tindakan hukum terhadap varian lebih sulit. Inilah kejahatan

yang telah ada selama berabad-abad di dunia *offline*. Penipuan, pencurian, dan sejenisnya sudah ada bahkan sebelum pengembangan peralatan berteknologi tinggi. Penjahat yang sama hanya diberi alat yang meningkatkan potensi korbannya dan membuatnya semakin sulit dilacak dan ditangkap. Kejahatan yang menggunakan jaringan komputer atau perangkat untuk memajukan tujuan lainnya meliputi:

- Penipuan dan pencurian identitas, walaupun semakin banyak menggunakan perangkat lunak perusak, peretasan dan atau *phishing*, menjadikannya sebagai contoh kejahatan komputer sebagai sasaran dan komputer sebagai alat;
- Perang informasi;
- Penipuan *phishing*;
- Spam;
- Konten cabul atau ofensif ilegal, termasuk pelecehan dan ancaman.

3.3 *Vulnerability*

Dalam keamanan komputer, *vulnerability* (kerentanan) adalah kelemahan yang memungkinkan penyerang untuk mengurangi jaminan informasi suatu sistem. Kerentanan adalah perpotongan tiga elemen: kerentanan atau cacat sistem, akses penyerang terhadap kekurangan, dan kemampuan penyerang untuk memanfaatkan kekurangan tersebut. Untuk memanfaatkan kerentanan, penyerang harus memiliki setidaknya satu alat atau teknik yang sesuai yang dapat terhubung ke kelemahan sistem. Dalam kerangka ini, kerentanan juga dikenal sebagai serangan permukaan.



Gambar 3.4 Siklus *Vulnerability Management*
(Lehtinen & Sr, n.d.)

Praktik siklus *vulnerability management* seperti terlihat pada Gambar 3.4 adalah mengidentifikasi, mengklasifikasi, memperbaiki, dan mengurangi kerentanan. Praktik ini umumnya mengacu pada kerentanan perangkat lunak dalam sistem komputasi. Risiko keamanan dapat diklasifikasikan sebagai kerentanan. Penggunaan kerentanan dengan arti risiko yang sama bisa menimbulkan kebingungan. Risiko itu terkait dengan potensi kerugian yang signifikan. Lalu ada kerentanan tanpa risiko, misalnya saat aset yang terkena dampak tidak memiliki nilai. Kerentanan dengan satu atau lebih contoh yang diketahui dari serangan yang dilakukan dan yang sepenuhnya dilaksanakan diklasifikasikan sebagai kerentanan yang dapat dieksploitasi. Bug keamanan (*security defect*) adalah konsep yang lebih sempit. Ada kerentanan yang terkait dengan perangkat lunak, perangkat keras, dan situs adalah contoh kerentanan yang merupakan keamanan bug. Konstruksinya dalam bahasa pemrograman bisa menjadi sumber kerentanan yang besar karena bisa dimanfaatkan oleh satu atau lebih penyerang di mana aset bernilai bagi organisasi, operasi bisnis kontinuitasnya, dan termasuk sumber informasi yang mendukung misi organisasi.

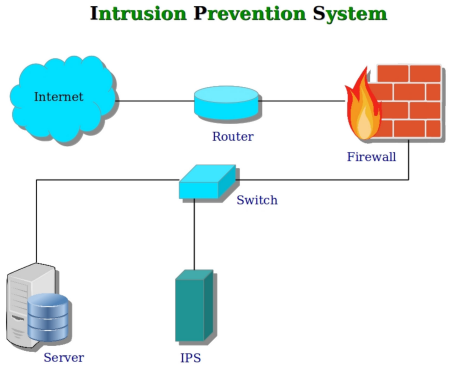
3.4 *Network Security*

Network security (keamanan jaringan) terdiri dari kebijakan yang diadopsi untuk mencegah dan memantau akses, penyalahgunaan, modifikasi, atau penolakan jaringan komputer dan sumber daya yang dapat diakses oleh jaringan oleh pihak yang tidak sah. Keamanan jaringan melibatkan otorisasi akses terhadap data dalam jaringan, yang dikendalikan oleh administrator jaringan. Pengguna memilih atau diberi ID dan kata sandi atau informasi autentikasi lainnya yang memungkinkan mereka mengakses informasi dan program sesuai kewenangannya. Keamanan jaringan mencakup berbagai jaringan komputer, baik publik maupun swasta, yang digunakan dalam pekerjaan sehari-hari seperti melakukan transaksi, komunikasi antarpelaku usaha, instansi pemerintah dan perorangan. Jaringan bisa bersifat pribadi, seperti di dalam perusahaan, dan lainnya yang mungkin terbuka untuk akses publik. Keamanan jaringan terlibat dalam organisasi, perusahaan, dan jenis institusi lainnya. Mengamankan jaringan, sekaligus melindungi dan mengawasi operasi yang dilakukan dengan cara yang paling umum dan sederhana untuk melindungi sumber daya jaringan adalah dengan nama yang unik (*username*) dan kata kunci (*password*) yang sesuai.

Konsep keamanan jaringan dimulai dengan autentikasi, biasanya dengan *username* dan *password*. Karena ini hanya memerlukan satu detail untuk mengautentikasi nama pengguna, yaitu kata sandi: ini terkadang disebut autentikasi satu faktor. Dengan autentikasi dua faktor, sesuatu yang dimiliki pengguna juga digunakan, misalnya token keamanan atau *'dongle'*, kartu ATM, atau ponsel, dan dengan autentikasi tiga faktor, sesuatu yang digunakan pengguna juga digunakan misalnya sidik jari atau pemindaian retina.

Setelah dikonfirmasi, *firewall* memberlakukan kebijakan akses seperti layanan apa yang diizinkan diakses oleh pengguna jaringan. Meskipun efektif untuk mencegah akses yang tidak sah, komponen ini

mungkin gagal memeriksa konten yang berpotensi berbahaya seperti *worm* komputer atau *trojan* yang dikirim melalui jaringan. Perangkat lunak anti-virus atau *intrusion prevention system* (IPS) seperti telihat pada Gambar 3.5 membantu mendeteksi dan menghambat serangan *malware*. *Intrusion prevention system* berbasis anomali juga dapat memantau jaringan seperti lalu lintas *wireshark* dan dapat dicatat untuk tujuan audit dan untuk analisis tingkat tinggi selanjutnya.



Gambar 3.5 Cara Kerja Anti-virus atau *Intrusion Prevention System* (IPS)

Komunikasi antara dua *host* menggunakan jaringan dapat dienkripsi untuk menjaga privasi. *Honeypots*, yang pada dasarnya memikat sumber daya yang dapat diakses oleh jaringan, digunakan di dalam jaringan sebagai alat pengawasan dan peringatan dini untuk menghindari akses untuk tujuan yang tidak sah. Teknik *honeypots* digunakan untuk merekam aktivitas penyerang yang mencoba untuk mengakses ke dalam suatu sistem, hasil rekaman aktivitas penyerang dipelajari untuk mengenal teknik eksploitasi baru. Analisis semacam itu dapat digunakan untuk lebih memperketat keamanan jaringan sebenarnya yang dilindungi oleh *honeypot*. Sebuah *honeypot* juga bisa mengarahkan perhatian penyerang dari server yang sah. Sebuah *honeypot* mendorong penyerang menghabiskan waktu dan energi mereka di server palsu

sambil mengalihkan perhatian mereka dari data pada server sebenarnya.

Mirip dengan *honeypo*t, *honeynet* adalah jaringan yang disiapkan dengan kerentanan yang disengaja. Tujuannya juga untuk mengundang serangan sehingga metode penyerang bisa dipelajari dan informasi itu bisa digunakan untuk meningkatkan keamanan jaringan. *Honeynet* biasanya mengandung satu atau lebih *honeypots*.

Manajemen keamanan untuk jaringan berbeda untuk semua jenis dan situasi. Rumah atau kantor kecil mungkin hanya memerlukan keamanan dasar sementara bisnis besar mungkin memerlukan perangkat lunak dan perangkat lunak dengan tingkat pemeliharaan tinggi dan lanjutan untuk mencegah serangan berbahaya dari *hacking* dan *spamming*. Jenis serangan dibagi menjadi dua kategori:

1. Pasif

   Ketika penyusup jaringan mencegat lalu lintas data yang berjalan melalui jaringan. Contoh serangan pasif adalah:

   - *Wiretapping*
   - *Port scanner*

2. Aktif

   Di mana penyusup memulai sebuah atau beberapa perintah untuk mengganggu operasi normal jaringan. Contoh serangan aktif adalah:

   - *Denial-of-service attack*
   - *DNS spoofing*
   - *Man in the middle*
   - *ARP poisoning*
   - *VLAN_hopping*
   - *Smurf attack*
   - *Buffer overflow*

- *Heap overflow*
- *Format string attack*
- *SQL injection*
- *Phishing*
- *Cross-site scripting*
- *CSRF*
- *Cyber-attack*

3.5 Soal Latihan

1) Jelaskan definisi dari internet *security*!
2) Jelaskan secara singkat cara kerja dari *firewall* dan tiga jenis *firewall*!
3) Sebutkan dan jelaskan tipe-tipe dari *Malware*!
4) Jelaskan secara singkat dari siklus *vulnerability management*!
5) Sebutkan dan jelaskan dua kategori jenis serangan!

Bab 4
Network Forensics Tools

Tujuan Instruksional Umum:

1. Mahasiswa/wi mampu menjelaskan tentang *network forensics tools*.

Tujuan Instruksional Khusus:

1. Mahasiswa/wi mampu memahami tentang *network forensic analysis tools* (NFAT).
2. Mahasiswa/wi mampu memahami tentang *vulnerability assessment*.
3. Mahasiswa/wi mampu memahami tentang *intrusion detection system* (IDS).
4. Mahasiswa/wi mampu memahami tentang *hardware*.
5. Mahasiswa/wi mampu memahami tentang konsep bukti digital.

4.1 Pendahuluan

Alat *network forensics* memungkinkan kita untuk memantau jaringan, mengumpulkan informasi tentang lalu lintas, dan membantu penyelidikan kejahatan jaringan. Alat forensik membantu dalam menganalisis penyusup, penyalahgunaan sumber daya, memprediksi

target serangan, melakukan penilaian risiko, mengevaluasi kinerja jaringan, dan melindungi hak kekayaan intelektual. Alat forensik dapat menangkap lalu lintas jaringan secara keseluruhan, memungkinkan pengguna menganalisis lalu lintas jaringan sesuai kebutuhan mereka dan menemukan fitur penting tentang lalu lintas data.

Ada beberapa alat forensik yang tersedia di pasaran, baik itu yang bersifat komersial atau *open source*, dalam perolehan data mempunyai keandalan dan kemampuan analisis yang kuat. Ada banyak pula perangkat *security and monitoring* (NSM) *open-source* lainnya yang dikembangkan hanya difungsikan untuk keamanan jaringan seperti *firewall* atau IDS, mereka tidak dirancang untuk pengumpulan barang bukti dan analisis tingkat lanjut. Namun, mereka dapat digunakan untuk membantu aktivitas analisis forensik tertentu.

Ada lima kategori *security and monitoring* (NSM) berdasarkan fungsinya, yaitu: *network forensic analysis tools* (NFAT), *vulnerability assessment, scanning, sniffing, analysis open source tool,* dan IDS. Berikut penjelasan lengkap dari lima *security and monitoring* (NSM) berdasarkan fungsinya:

4.2 *Network Forensic Analysis Tools* (NFAT)

Network forensic analysis tools (NFAT) bersifat *proprietary* dan *open source*. Di mana, alat *proprietary* dibangun untuk *logging*, pencatatan dan penyimpanan data jaringan sebagai bukti evaluasi. *Logged* data dapat disimpan lebih dari satu tahun untuk menyelidiki kejadian jaringan yang berbasis waktu. Perangkat kebanyakan perangkat lunak, dibangun untuk lingkungan Linux yang dengan mudah diprogram ulang untuk menambahkan fungsi tambahan lainnya. Mereka juga bisa dikuasai dan juga bisa digabungkan menjadi alat baru dan hebat.

NetDetektor adalah alat forensik berfitur lengkap yang dibangun di atas arsitektur Alpine NIKSUN. Ini mengintegrasikan IDS berbasis

signature-based dengan deteksi anomali statistik dengan rekonstruksi aplikasi lengkap, dekode tingkat paket, dan lain-lain. NetDetector menginformasikan pengguna tentang pelanggaran keamanan dan dapat melakukan tindakan pencegahan seperti memblokir lalu lintas berbahaya memasuki sistem. Beberapa fitur utama dari NetDetector adalah sebagai berikut:

- Keamanan dan kecerdasan data yang besar.
- Rekonstruksi aplikasi dan sesi.
- IDS berbasis tanda tangan terpadu dan deteksi anomali.
- Penangkapan lalu lintas dan analisis multi-skala waktu.
- Pelaporan *ad hoc* dan terjadwal.

NetIntercept sebuah alat terintegrasi dengan sistem yang lengkap dengan perangkat keras dan perangkat lunak pra-instal, siap untuk solusi forensik. NetIntercept dapat ditempatkan di perangkat tersendiri atau di *firewall*. NetIntercept mencatat lalu lintas di *hard disk*. Oleh karena itu, lalu lintas dari jam, hari, atau minggu terakhir tersedia untuk analisis. Analisis umumnya dianalisis dalam mode *batch*.

NetIntercept mengorelasikan sesi pengguna dan merekonstruksi *file* yang dikirim atau diterima, memberikan bukti langsung tentang tindakan kejahatan di jaringan. NetIntercept bertujuan untuk menjawab pertanyaan dasar sebagai berikut ini:

- Siapa yang mengirimkan informasi?
- Mengapa informasi tidak bergerak?
- Bagaimana sistem diserang?

Seperti terlihat pada Gambar 4.1, menyediakan visibilitas *real-time* ke setiap bagian jaringan. Ini memiliki kemampuan untuk menangkap paket yang tinggi, konsol terpusat, mesin terdistribusi, dan analisis. OmniPeek mendukung Ethernet, Gigabit, 10 Gigabit, 802.11a / b / g / n / ac wireless, VoIP, video, MPLS, dan VLAN. OmniPeek tersedia

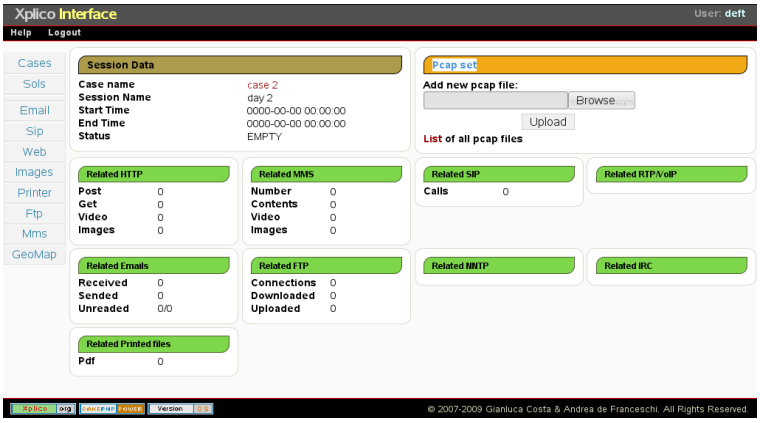
dalam empat versi: dasar, profesional, perusahaan, dan terhubung. Setiap versi menyediakan beberapa fitur unik dan mendukung berbagai jenis jaringan.

Gambar 4.1 OmniPeek

Python Forensic Log Analysis GUI (PyFLAG)) adalah alat analisis forensik dan log *database* berbasis web yang ditulis dengan *python*. PyFlag menganalisis paket yang ditangkap dalam format *libpcap* dan mendukung sejumlah protokol jaringan. Memiliki kemampuan untuk memeriksa data secara *rekursif* pada berbagai tingkatan dan sangat sesuai untuk protokol jaringan yang biasanya berlapis. PyFlag memilah-milah *file* pcap, mengekstrak paket, dan membedahnya pada protokol *lowlevel* (IP, TCP, atau UDP). Paket terkait dikumpulkan ke dalam arus menggunakan *reassembler*. Aliran ini kemudian dibedah dengan protokol di sektor tingkat tinggi seperti HTTP dan IRC.

Xplico, seperti terlihat pada Gambar 4.2 adalah alat analisis forensik *open-source* untuk sistem UNIX. Xplico mampu merekonstruksi data aplikasi protokol dari paket yang diambil. Xplico secara khusus dibuat untuk rekonstruksi data protokol. Ini menggunakan teknik bernama *port independent protocol identification* (PIPI) untuk mengenali protokol. Xplico membedah data pada tingkat protokol dan merekonstruksi dan

menormalkannya untuk digunakan dalam manipulator. Para manipulator kemudian men-*transkode*, mengorelasikan, dan mengumpulkan data untuk dianalisis dan menyajikan hasilnya dalam bentuk visualisasi.



Gambar 4.2 Xplico Interface

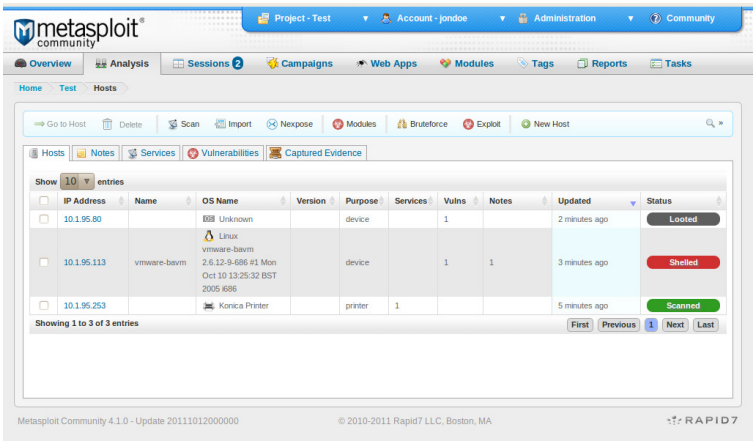
4.3 *Vulnerability Assessment*

Vulnerability/kerentanan merupakan bagian integral yang melekat dari sistem berbasis komputer baik perangkat lunak maupun perangkat keras. *Bug* dalam perangkat lunak komersial atau celah dalam sistem, *bug* di sistem operasi, *misconfigurations* membuat sistem rentan terhadap serangan atau akses berbahaya. Orang jahat dapat mengakses sistem menggunakan celah dan *bug* ini. Dari sudut pandang teknis, usaha semacam itu tidak mudah, tapi ada beberapa insiden di masa lalu yang memiliki reputasi kerentanan yang diketahui dan tidak dikenal dapat dimanfaatkan oleh pengguna jahat serta orang atau tidak sah baik dari dalam maupun luar organisasi.

Alat *vulnerability assessment* bertujuan untuk menemukan kerentanan dalam suatu sistem. Terkadang, alat *vulnerability assessment* memindai sistem kerentanan yang diketahui dan terkadang menutupi

serangan palsu untuk menemukan kerentanan baru. Beberapa alat populer diberikan di bawah ini.

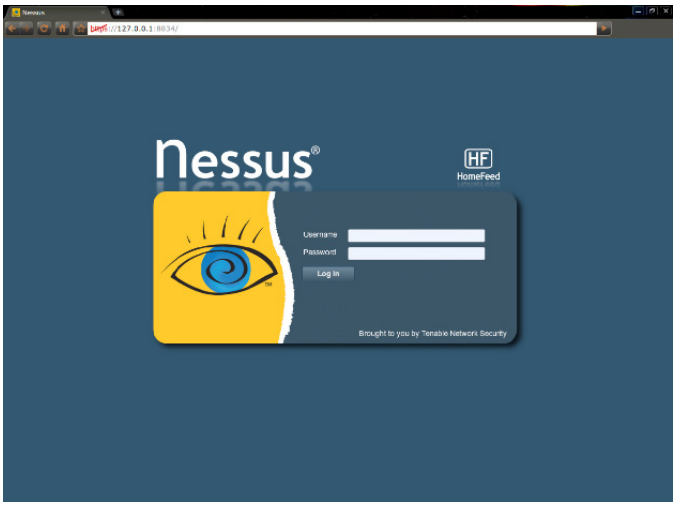
Metasploit yang ditunjukkan pada Gambar 4.3 pada dasarnya adalah kerangka pengujian penetrasi. *Metasploit* menyediakan platform "eksploitasi" untuk pengembangan, pengujian, dan penggunaan untuk memanfaatkan kerentanan sistem. Awalnya *metasploit* adalah *open source* tapi Rapid7 mengakuisisinya pada tahun 2009, tersedia untuk Windows, Linux, dan MAC. *Metasploit* tersedia dalam tiga versi: *metasploit community edition* bersifat gratis namun memiliki keterbatasan fungsionalitas, *metasploit express* dengan beberapa fitur canggih tambahan, dan *metasploit pro* merupakan fitur dengan fitur lengkap. *Metasploit framework* masih *open source* dan tersedia untuk diunduh.



Gambar 4.3 *Metasploit*

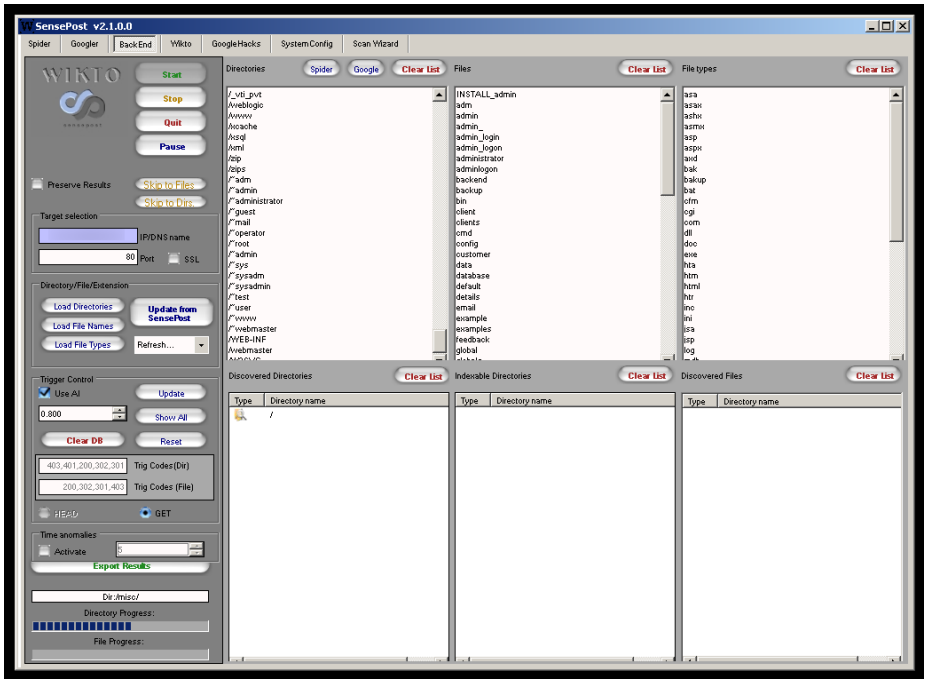
Nessus yang ditunjukkan di Gambar 4.4 adalah alat pemindaian kerentanan untuk Windows, Linux, Solaris, FreeBSD, dan MAC. Nessus dimiliki oleh *tenable network security*, tersedia dalam empat versi, evaluasi Nessus, Nessus, Nessus Perimeter Service, dan Nessus Home. Nessus menawarkan pemeriksaan kerentanan baru dalam bentuk *plug-*

in setiap hari. Hal ini juga digunakan untuk *scanning* serangan DoS, *port scan*, dan *password vulnerabilities*.



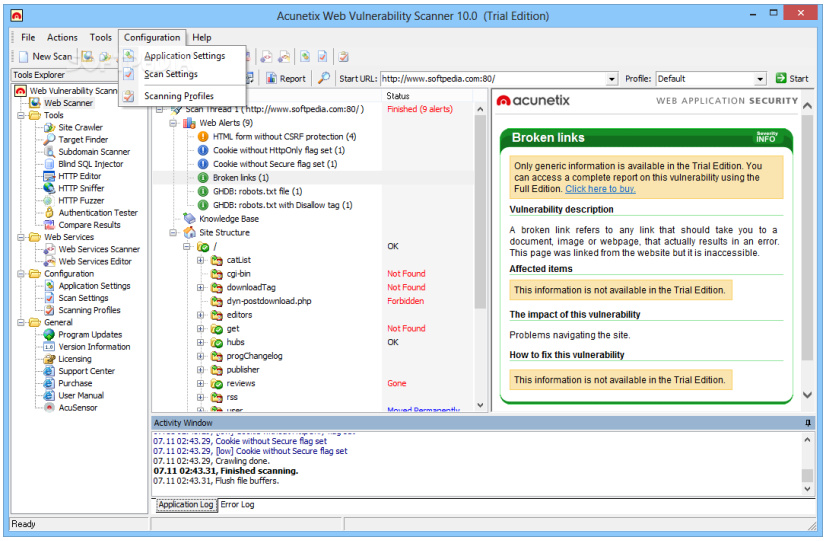
Gambar 4.4 Nessus

Wikto seperti ditunjukkan pada Gambar 4.5, dan Nikto adalah alat *assessment* web server yang bertujuan untuk memeriksa kekurangan di server web. Wikto memindai web server untuk menemukan direktori dan *file* yang tersimpan di server tersebut. Ini terlihat untuk diketahui kerentanan dan skrip lainnya yang dapat disalahgunakan atau dieksploitasi dalam implementasi server. Wikto dapat dilihat sebagai Nikto untuk Windows, hanya dengan beberapa fitur tambahan seperti pengecekan kode logika berbasis *fuzzy*, kemampuan penambangan *back-end*, penambangan direktori yang dibantu oleh Google, dan pemantauan permintaan/tanggapan HTTP *real-time*.



Gambar 4.5 *Wikto Interface*

Acunetix web vulnerability scanner adalah aplikasi yang secara otomatis memeriksa keseluruhan kerentanan keamanan dari situs web atau aplikasi web. *Acunetix web vulnerability scanner* mampu memindai seluruh kode dan skrip yang kemungkinan kerentanan yang dapat dieksploitasi. Ini juga mencakup beberapa alat pengujian penetrasi untuk mengotomatisasi keseluruhan proses, serta menciptakan serangan palsu dan memeriksa respons situs web terhadap serangan tersebut.



Gambar 4.6 *Acunetix Web Vulnerability Scanner*

Beberapa fitur Acunetix WVS dipakai untuk pengujian injeksi SQL dan uji coba lalu lintas situs, dukungan untuk halaman CAPTCHA, pemindaian *multithread*, memindai *port* web server dan menjalankan pemeriksaan keamanan terhadap layanan jaringan yang berjalan di server, dan alat uji penetrasi yang canggih seperti: editor HTTP dan HTTP fuzzer. Setelah pemindaian selesai, *Acunetix web vulnerability scanner* akan menampilkan laporan rinci tentang apa yang telah ditemukan dan bagaimana memperbaiki keamanan seperti ditunjukkan pada Gambar 4.6.

4.4 *Network Sniffing dan Packet Analyzing Tools*

Sniffing dan *packet analyzing tools* meliputi perangkat lunak atau perangkat keras yang dapat mencegat dan menangkap paket data yang melewati jaringan atau segmen jaringan. *Sniffer* menangkap paket data dan mampu melakukan *decoding* dan menampilkan berbagai bidang paket. Alat analisis paket digunakan untuk menganalisis paket

yang diambil berdasarkan RFC atau standar lainnya. Mengendus dan menganalisis alat membantu dalam menganalisis masalah jaringan, mendeteksi upaya eksploitasi yang mengisolasi sistem yang dieksploitasi, dan penggunaan sistem pemantauan.

Libpcap dan WinPcap adalah perangkat lunak bebas yang dirilis di bawah lisensi BSD, yang telah disetujui oleh *open source initiative*. Libpcap adalah *library* UNIX C yang menyediakan API untuk menangkap dan menyaring *frame linklayer* data dari antarmuka jaringan. Awalnya dikembangkan di Lawrence Berkeley National Laboratory (LBNL) dan dirilis ke publik pada bulan Juni 1994. Sistem UNIX yang berbeda memiliki arsitektur yang berbeda untuk memproses *frame link-layer*. Tujuan dari libpcap adalah untuk menyediakan lapisan abstraksi sehingga pemrogram dapat merancang alat pengambilan dan analisis paket portabel.

Pada tahun 1999, *Computer Networks Group* (NetGroup) di Politecnico di Torino menerbitkan WinPcap , sebuah *library* berdasarkan libpcap yang dirancang untuk sistem Windows. Sejak saat itu, banyak orang dan perusahaan telah berkontribusi pada proyek WinPcap. Kode ini sekarang *dihosting* di situs yang dikelola oleh Riverbed Technology.

Alat perekam dan analisis paket populer saat ini didasarkan pada *library* libpcap Ini termasuk tcpdump, Wireshark, Snort, nmap, ngrep, dan banyak lainnya. Akibatnya, alat ini dapat dioperasikan dalam arti bahwa paket yang ditangkap dengan satu alat dapat dibaca dan dianalisis dengan yang lain. Fitur klasik dari utilitas berbasis libpcap adalah mereka dapat menangkap paket di *layer 2* dari hampir semua perangkat antarmuka jaringan dan menyimpannya dalam *file* untuk analisis selanjutnya. Alat lain kemudian dapat membaca di *file* "*packet capture*" dan atau disingkat dengan pcap. Banyak alat yang berbasis pada libpcap juga mencakup fungsionalitas khusus, seperti kemampuan untuk menggabungkan tangkapan paket, yaitu *mergecap*, untuk membagi aliran

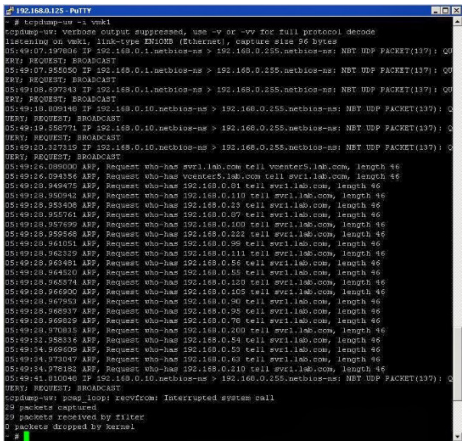
TCP yaitu tcpflow, atau untuk melakukan pencarian ekspresi reguler pada isi paket misalnya *ngrep*.

Di era modern, volume data yang mengalir melintasi jaringan begitu besar sehingga sangat penting bagi penyidik untuk mengetahui bagaimana menyaringnya selama pengambilan dan analisis. Libpcap mencakup bahasa penyaringan yang sangat kuat yang disebut *sintaks* Berkeley Packet Filter (BPF). Menggunakan filter BPF, penyidik dapat memutuskan lalu lintas mana yang akan ditangkap dan diperiksa dan lalu lintas mana yang harus diabaikan. BPF memungkinkan memfilter lalu lintas berdasarkan perbandingan nilai di bidang untuk protokol *layer 2, 3,* dan *4*. Filter BPF bisa terdiri dari rantai kondisional yang rumit, AND dan OR. *Sintaks* BPF sangat banyak digunakan dan didukung oleh alat analisis lalu lintas dan analisis yang setiap penyidik jaringan harus mengenalinya.

Tcpdump adalah perangkat lunak/alat untuk menangkap, memfilter, dan menganalisis lalu lintas jaringan. Dikembangkan oleh BNL dan pertama kali dirilis ke publik pada bulan Januari 1991. Meskipun tcpdump saat ini bergantung pada libpcap untuk fungsionalitas, rilis publiknya benar-benar mendahului libpcap. Tcpdump dirancang sebagai alat UNIX. Pada tahun 1999, para peneliti di Politecnico di Torino memasukkannya ke Windows dan merilis WinDump untuk versi Windows. WinDump sekarang *dihosting* di situs yang dikelola oleh Riverbed Technology, bersama dengan WinPcap. Tcpdump dan utilitas WinDump tidak sepenuhnya identik, tetapi biasanya menghasilkan hasil yang sebanding.

Packet capture yang diproduksi oleh WinDump tidak dapat dibaca oleh tcpdump, namun kejadian seperti itu jarang terjadi. Tujuan dasar tcpdump adalah untuk menangkap lalu lintas jaringan dan kemudian mencetak atau menyimpan konten untuk di analisis. Tcpdump menangkap lalu lintas sedikit demi sedikit saat melintasi media fisik

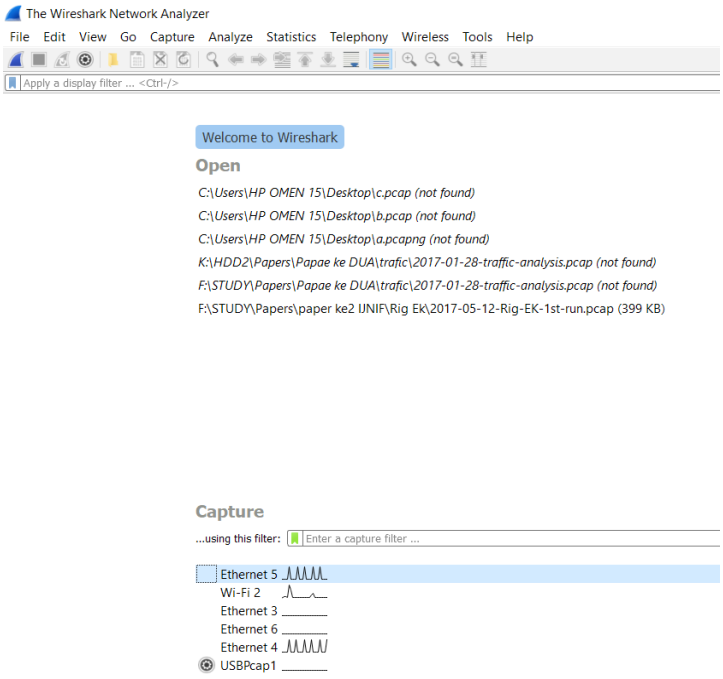
pada trafik *link-layer* baik di kabel tembaga, serat , atau bahkan *wireless*, seperti terlihat pada Gambar 4.7



Gambar 4.7 Tcpdump

Karena tcpdump didasarkan pada libpcap, ia menangkap *layer 2*/data *link layer*, secara *default*, *output* tcpdump hanya menampilkan rincian protokol *layer 3* dan di atas, namun dapat digunakan untuk menampilkan data *layer 2* juga. Di luar sekadar menangkap paket, tcpdump dapat memecahkan kode protokol *layer 2* sampai *4* yang umum dan beberapa protokol lapisan tinggi juga. Kemudian menampilkan informasi paket yang didekode dan ditampilkan dalam heksadesimal atau ASCII.

Wireshark adalah perangkat lunak *open source*, bertampilan antarmuka grafis yang dirancang untuk menangkap, memfilter, dan menganalisis lalu lintas. Antarmuka grafis Wireshark yang mudah, menjadikannya alat pertama dan utama yang hebat untuk analisis *network forensics*, serta mempunyai kemampuan penyaringan paket tingkat lanjut, fitur *decoding* protokol, dan dukungan untuk *packet details markup language* (PDML) yang juga membuat sangat berguna untuk penyidik pemula atau yang sudah berpengalaman.



Gambar 4.8 *Panel Capture Options Wireshark*

Wireshark awalnya bernama Ethereal, dirilis pada tahun 1998 oleh Gerald Combs. Lantas, nama itu diubah pada tahun 2006 ketika Combs dipindahkan ke CACE Technologies karena perusahaan sebelumnya tetap mempertahankan merek dagang Ethereal. Selanjutnya, CACE Technologies diakuisisi oleh Riverbed Technology. Seiring waktu ada ratusan kontributor dalam pengembangan Wireshark. Wireshark, seperti terlihat pada Gambar 4.8, memungkinkan menangkap paket pada setiap sistem antarmuka jaringan dengan asumsi memiliki izin yang sesuai untuk melakukannya dan kartu jaringan telah mendukung *sniffing*. Wireshark dapat menampilkan paket saat mereka ditangkap, tepat saat itu juga serta analisis protokol yang sangat kuat dengan menggunakan banyak kekuatan pemrosesan untuk mengelolah data protokol.

Fitur utama Wireshark meliputi:

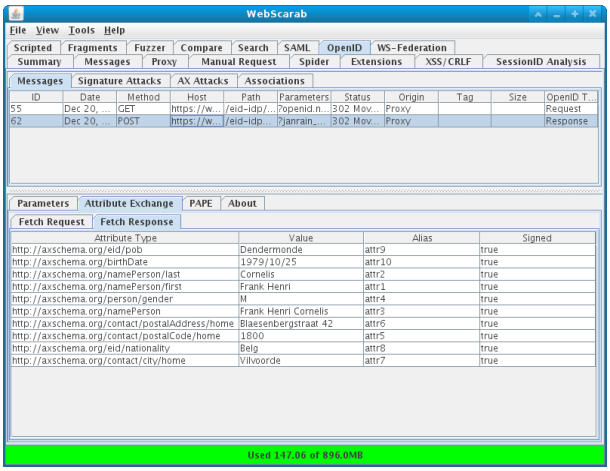
- Dukungan untuk Ethernet, IEEE 802.11, PPP, *loopback* dan USB.
- Analisis *live capture* dan *offline*.
- GUI interaktif dan juga versi *command line*.
- *Plug-in* dapat dibuat untuk menganalisis protokol baru.
- Menyediakan analisis VoIP.
- *Output* dapat diekspor ke sejumlah format *file,* seperti XML, CSV, plain text, dan PostScript.

Gambar 4.9 Aircrack-ng

Aircrack-ng seperti ditunjukkan pada Gambar 4.9 di atas adalah perangkat lunak jaringan untuk jaringan lokal berbasis nirkabel IEEE 802.11. Terutama terdiri dari detektor, *packet sniffer*, WPA/WPA2-PSK, alat *cracking* dan analisis WEP. Aircrack-ng tersedia untuk Windows dan Linux dan bekerja dengan kartu *network interface card* (NIC) yang mendukung mode pemantauan.

WebScarab ditulis dengan mengguanakan *Java* dan menganalisis aplikasi yang menggunakan protokol HTTP dan HTTPS untuk komunikasi. WebScarab beroperasi sebagai perantara antara internet

dan aplikasi yang memungkinkan untuk meninjau dan memodifikasi permintaan keluar dan masuk, seperti ditunjukkan pada Gambar 4.10.



Gambar 4.10 WebScarab

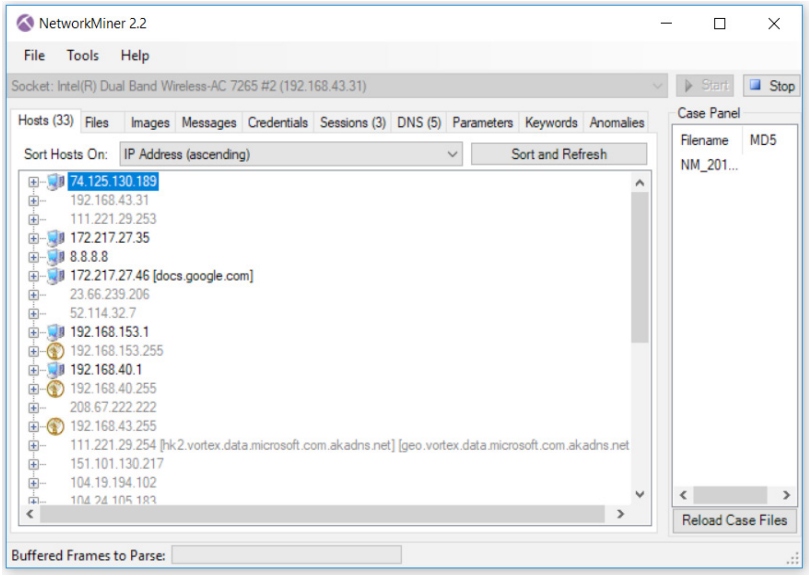
WebScarab memiliki beberapa fitur berupa *plug-in*, seperti:

- *Proxy*: Untuk mengamati lalu lintas antara aplikasi dan web.
- *Intercept manual*: Memungkinkan pengguna untuk memodifikasi, meminta, dan menanggapi HTTP/HTTPS sebelum mencapai server.
- Permintaan manual: Memungkinkan pengeditan manual atau pengulangan permintaan sebelumnya atau untuk membuat permintaan yang sama sekali baru.
- *Bandwidth emulator*: Memungkinkan pengguna untuk meniru jaringan yang lebih lambat.

Ngrep adalah penganalisis paket jaringan sumber terbuka untuk Linux dan dapat diimpor ke sistem operasi UNIX lainnya. *Ngrep* dapat mencari lalu lintas yang berasal dari *port* tertentu dan dapat mencari ekspresi reguler dalam paket *payload*. *Ngrep* memungkinkan pengguna melihat semua lalu lintas yang tidak dienkripsi pada jaringan dan

mendukung sejumlah protokol: IPv4 dan IPv6, TCP, UDP, ICMPv4 dan ICMPv6, IGMP, SLIP, PPP, FDDI, Ethernet, dan lain-lain.

NetworkMiner adalah *sniffing tool* pada forensik jaringan untuk sistem operasi berbasis Windows. NetworkMiner seperti ditunjukkan pada Gambar 4.11 dapat mendeteksi IP, nama *host*, sistem operasi, dan *port* terbuka. Hal ini juga dapat mengekstrak *file* yang dikirim melalui jaringan, mengumpulkan data tentang *host* dan mengumpulkan lalu lintas data.

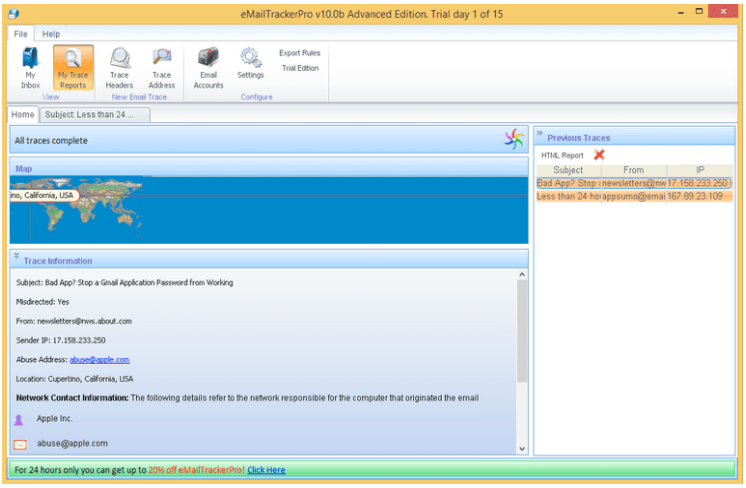


Gambar 4.11 *NetworkMiner*

Kismet adalah detektor jaringan nirkabel dan *sniffer*. Kismet bekerja dengan adaptor nirkabel yang mendukung mode pemantauan/monitor mode (rfmon). Kismet seperti ditunjukkan pada Gambar 4.12, mampu mendeteksi jaringan dengan nama SSID serta jaringan tersembunyi dan *non-beaconing*. Kismet secara pasif mengumpulkan paket tanpa mengganggu lalu lintas jaringan.

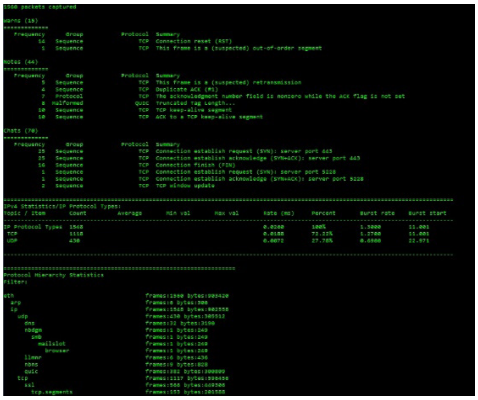
Gambar 4.12 Kismet

EMailTrackerPro menawarkan kemampuan untuk melacak *e-mail* menggunakan tajuk *id e-mail* sendiri ke lokasi asalnya. Versi lanjutan dari eMailTrackerPro juga dilengkapi dengan filter spam, yang memindai setiap *e-mail* masuk untuk dugaan sifat spam mereka. EMailTrackerPro dapat melacak beberapa IP dan domain sekaligus, seperti ditunjukkan pada Gambar 4.13.



Gambar 4.13 eMailTrackerPro

Tshark adalah alat analisis protokol jaringan berbasis baris perintah yang merupakan bagian dari distribusi Wireshark. Seperti Wireshark, ini berbasis libpcap, dan dapat membaca dan menyimpan *file* dalam format standar yang sama dengan Wireshark. Selain analisis, tshark juga bisa digunakan untuk menangkap paket. Contoh di Gambar 4.14 di bawah ini menunjukkan tshark menangkap lalu lintas pada antarmuka jaringan *eth0*, menyaring semua lalu lintas *por*t dan menyimpan hasilnya dalam *file pcap*.

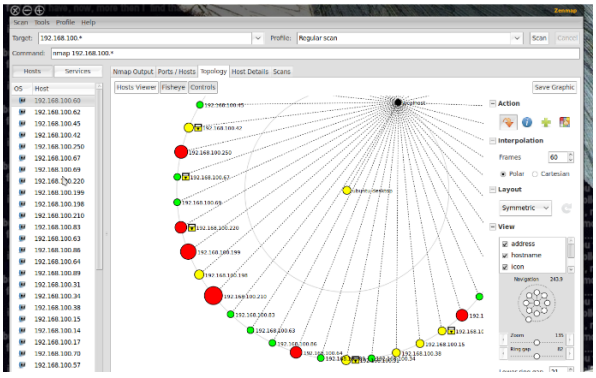


Gambar 4.14 *Tshark Network Scanning Tools*

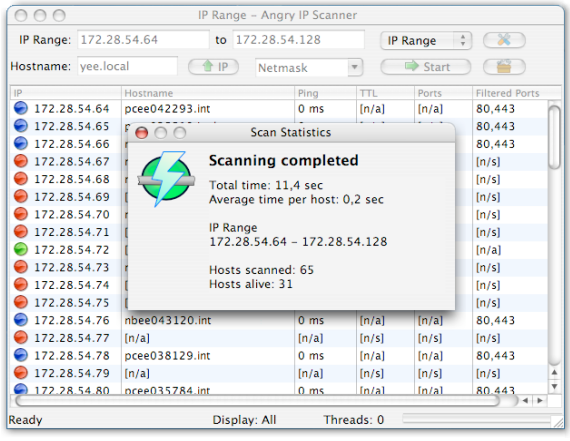
Network scanning adalah prosedur untuk menemukan *host* aktif dan layanan yang mereka tawarkan di dalam jaringan. Prosedur dasar *scanning* atau pemindaian adalah melibatkan *ping sweep* dan *port scan*. *Ping sweep* menerima kembali informasi tentang *host* aktif di jaringan sedangkan *port scan* mengembalikan layanan yang ditawarkan oleh *host*. Kadang-kadang prosedur pemetaan juga dapat digunakan untuk menemukan IP mana yang tidak sesuai dengan *host* aktif. Alat pemindaian jaringan menyediakan cara otomatis dan efisien untuk melaksanakan tugas ini.

Nmap (*Network Mapper*) adalah alat pemindai *port*, bersifat *open source* dan *cross platform*. Nmap berguna untuk memetakan IP ke

informasi *host* dan juga menawarkan rincian tentang layanan yang diberikan *host* atas jawaban pertanyaan, seperti: OS apa yang sedang digunakan? Atau jenis *packet filter/firewall* apa yang sedang digunakan? Dan lain lain. Nmap dilengkapi dengan pilihan berikut: Zenmap dengan penampil GUI yang ditunjukkan pada Gambar 4.15; Ncat: alat pengalih data dan debugging; Ndiff: utilitas untuk membandingkan hasil pemindaian; dan Nping.



Gambar 4.15 Zenmap dengan penampil GUI



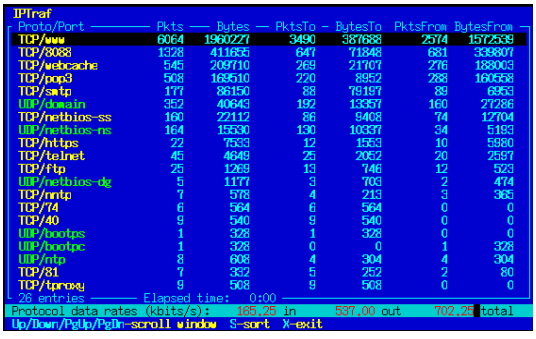
Gambar 4.16 *Angry IP Scanner*

Angry IP Scanner (atau IPScan) adalah *port scanner* dengan kemampuan *cross platform*, memindai alamat IP dalam rentang apa pun dengan melakukan *ping* ke mereka dan kemudian menampilkan nama *host*, alamat MAC, dan *port scan* yang aktif atau tidak. Hasil pemindaian, seperti ditunjukkan pada Gambar 4.16 dapat disimpan dalam berbagai format *file*, seperti *file* daftar CSV, TXT, XML, dan IP-Port. IP Scanner menggunakan pendekatan *multithread* untuk mempercepat proses *scanning*. Untuk setiap alamat IP, sebuah *thread* terpisah dibuat. Fungsi IP Scanner dapat diperpanjang dengan menggunakan *plug-in*.

4.5 *Network Monitoring Tools*

Jaringan komputer tidak dapat diandalkan maka diperlukan alat untuk membantu dan memantau kinerja jaringan, QOS, *delay*, dan *bandwidth*.

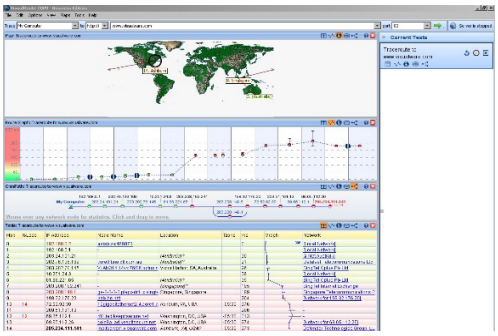
IPTraf adalah utilitas statistik jaringan berbasis konsol *open-source* berbasis UNIX. Ini mengumpulkan paket lalu lintas jaringan, seperti jumlah paket TCP, UDP, jumlah byte, aktivitas *host*, dan statistik antarmuka. IPTraf seperti ditunjukkan pada Gambar 4.17 mendukung sejumlah antarmuka seperti Ethernet, FDDI, ISDN, SLIP, PPP. IPTraf mendukung sejumlah jenis paket: IP, TCP, UDP, IGMP, ICMP, OSPF, IGP, ARP, dan RARP.



Gambar 4.17 *IPTraf*

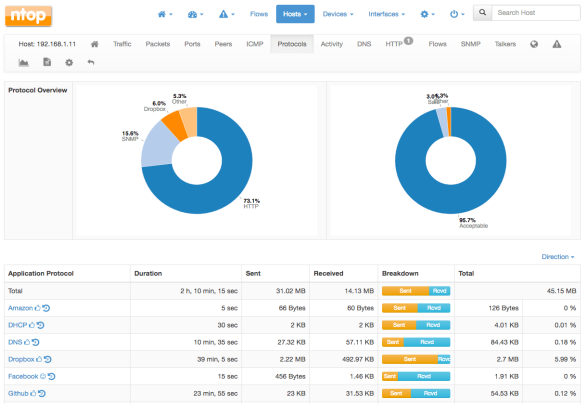
VisualRoute adalah alat diagnostik konektivitas yang menampilkan hasil *ping* dan *traceroute* dalam bentuk visual. VisualRoute digunakan terutama untuk mengatasi masalah konektivitas, evaluasi kinerja dari sebuah tautan berdasarkan *packet loss,* dan *latency.* VisualRoute menghasilkan *ping* interaktif dan informasi *whois* dengan grafik waktu. Hasil *trace* VisualRoute, seperti ditunjukkan pada Gambar 4.18 bisa diekspor ke teks, laporan HTML, atau *screenshot* JPG.

Fitur utama dari VisualRoute adalah sebagai berikut: pelaporan lokasi IP, pencarian *whois*, penemuan *multipath*, pengujian *port*, *port probing*, pengujian kinerja DNS, *traceroute* dan *reverse tracing*.



Gambar 4.18 VisualRoute

Ntop adalah alat yang menunjukkan statistik jaringan yang memiliki dua mode, mode interaktif dan mode web. Dalam mode interaktif, Ntop menunjukkan status jaringan di terminal. Dalam mode web, ia bertindak sebagai server web dan membuat dump HTML status jaringan dan menunjukkannya dalam bentuk halaman web, dengan grafik dan statistik lainnya seperti ditunjukkan pada Gambar 4.19. Ntop bisa mengurutkan lalu lintas sesuai dengan banyak protokol, menganalisis lalu lintas IP, dan mengurutkan menurut sumber atau tujuan.

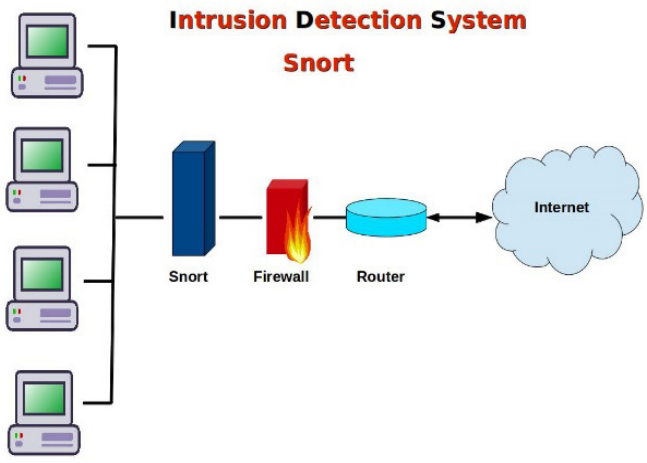


Gambar 4.19 Ntop

4.6 *Intrusion Detection System* (IDS)

Intrusion detection system adalah perangkat lunak atau perangkat keras yang memonitor aktivitas potensial berbahaya di jaringan. IDS bertujuan untuk mendeteksi lalu lintas dan aktivitas yang mencurigakan yang berasal dari dalam dan luar organisasi. IDS bisa berupa *network-based* (NIDS) atau *host-based* (HIDS). Bahkan beberapa IDS mungkin hanya mencoba menghentikan intrusi yang terdeteksi, namun kebanyakan IDS hanya bertujuan untuk mendeteksi dan melaporkan adanya gangguan.

Snort adalah *network intrusion detection system* (NIDS) atau deteksi sistem jaringan yang dirancang untuk jaringan berbasis IP. Snort menganalisis lalu lintas dan paket jaringan untuk mendeteksi *worm*, eksploitasi kerentanan, pemindaian *port*, dan perilaku mencurigakan lainnya. Aturan snort juga dapat didefinisikan oleh pengguna dan memeriksa berbagai atribut paket apakah lalu lintas harus diizinkan atau diblokir, seperti ditunjukkan pada Gambar 4.20.

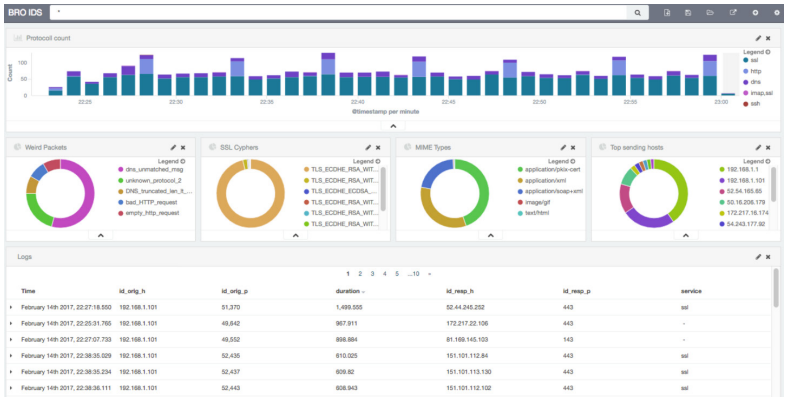


Gambar 4. 20 Snort

Snort bekerja dalam tiga mode:

1. *Mode sniffer*, hanya membaca paket jaringan dan *display* di konsol.
2. *Mode logger*, log dan menyimpan paket ke *disk*.
3. Mode deteksi intrusi, ia menganalisis lalu lintas jaringan terhadap kumpulan aturan yang ditetapkan.

Bro adalah NIDS pasif untuk sistem UNIX, monitor dan memindai lalu lintas jaringan yang mencurigakan secara mendalam. Bro mencatat aktivitas jaringan dalam *file log* dalam persyaratan tingkat tinggi. Bro dapat mencatat semua permintaan HTTP beserta URI, *header*, jenis MIME, permintaan DNS, sertifikat SSL, sesi SMTP, dan masih banyak lagi, seperti terlihat pada Gambar 4.21.



Gambar 4.21 Bro

Bro dapat dilihat sebagai platform untuk analisis lalu lintas dengan dilengkapi perpustakaan standar yang telah ditentukan dan mendukung berbagai fitur untuk mendeteksi gangguan.

4.7 *Hardware*

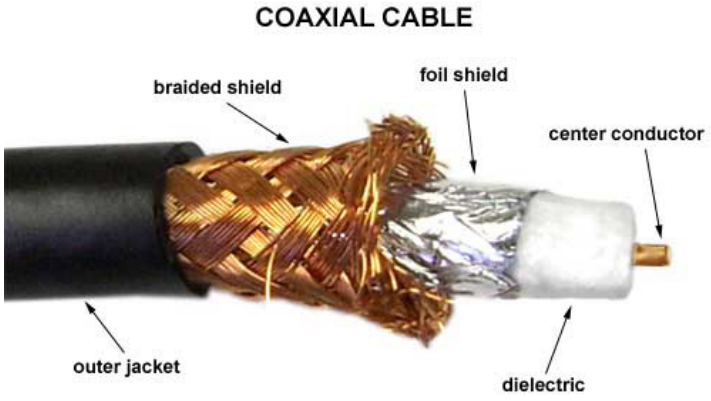
Idealnya, semua investigator atau investigator *network forensics* menginginkan *evidence* atau barang bukti yang sempurna, seperti kabel tembaga hanya untuk mengamati perubahan tegangan tanpa pernah memodifikasinya, kabel serat *optic* di mana hanya mengamati kuanta tanpa pernah menyuntikkan apa pun, frekuensi radio di mana hanya mengamati RF tanpa pernah memancarkan apa pun. Di dunia nyata, seorang investigator *network forensics* pembunuhan yang mengumpulkan bukti dari TKP tidak mungkin melakukan investigasi tanpa meninggalkan jejak baru. Jelas, tidak ada yang sempurna, dan tidak akan pernah bisa mencapai *zero footprint* atau tidak meninggalkan jejak. Investigator *network forensics* yang menganalisis kasus pembunuhan tidak bisa menghindari berjalan di atas lantai yang sama dengan si pembunuh.

Investigator *network forensics* sering mengacu pada akuisisi pasif versus aktif. Akuisisi bukti pasif adalah praktik mengumpulkan bukti forensik dari jaringan pada *layer 2* ke atas. Akuisisi lalu lintas data ini sering digolongkan sebagai akuisisi bukti pasif. Akuisisi bukti aktif atau interaktif adalah praktik pengumpulan bukti dengan berinteraksi dengan perangkat di jaringan. Ini seperti masuk ke perangkat jaringan melalui konsol atau melalui antarmuka jaringan, atau bahkan memindai *port* jaringan untuk menentukan keadaan saat ini. Meskipun istilah pasif dan aktif menyiratkan bahwa ada perbedaan yang jelas antara dua kategori tersebut, pada kenyataannya dampak akuisisi bukti terhadap lingkungan dari barang bukti *evidence* itu sendiri.

Kontak fisik hal ini dimungkinkan untuk mendapatkan lalu lintas jaringan tanpa mengirim atau memodifikasi *frame* data apa pun pada jaringan. Meskipun tidak mungkin untuk memiliki dampak nol terhadap lingkungan, tidak mungkinkan proses menangkap atau mengendus lalu lintas bisa dilakukan dengan dampak yang sangat kecil. Ada banyak cara untuk mentransmisikan data melalui media fisik, dan sama seperti banyak cara mencegat, yang paling sederhana adalah perangkat investigator yang terhubung ke perangkat lain melalui saluran fisik, seperti kabel UTP atau serat optik. Tegangan pada tembaga dapat dengan mudah diperkuat dan didistribusikan kembali dalam konfigurasi *satu-ke-banyak*. Hub dan *switch* dirancang untuk memperluas media fisik untuk berbagi *baseband* dengan perangkat tambahan. Penyelidik forensik secara pasif dapat memperoleh lalu lintas jaringan dengan cara mencegatnya seperti apa adanya ditransmisikan melintasi kabel, melalui udara, atau melalui peralatan jaringan seperti hub dan *switch*. Kabel memungkinkan koneksi *point-to-point* antarperangkat atau *host*. Bahan yang paling umum untuk kabel tembaga dan serat optik. Masing-masing bisa mengendus, meski peralatan dan efek sampingnya bervariasi berdasarkan media fisik.

Kabel tembaga memiliki dua jenis yang paling banyak digunakan, yaitu: kabel coaxial dan twisted pair:

1. Kabel coaxial, atau coax, seperti terlihat pada Gambar 4.22. Terdiri dari inti kawat tembaga tunggal yang dibungkus isolasi dan ditutup dengan perisai tembaga. Paket ini kemudian disegel dengan isolasi *outer*. Karena media transmisi adalah inti tembaga tunggal, semua stasiun di jaringan harus menegosiasikan transmisi dan penerimaan sinyal.

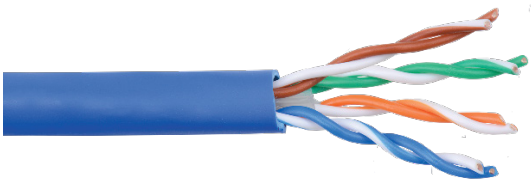


Gambar 4.22 Kabel Coaxial

Manfaatnya adalah bahwa inti tembaga terlindungi dari interferensi elektromagnetik. Dalam kebanyakan kasus di mana coax digunakan, jika Anda dapat menyentuh inti tembaga tunggal, Anda dapat mengakses lalu lintas ke dan dari semua *host* yang berbagi media fisik.

2. Kabel twisted pair (TP) seperti terlihat pada Gambar 4.23 mengandung banyak pasang kabel tembaga. Tidak seperti kabel koaksial, di mana inti tembaga tunggal terlindung dari gangguan elektromagnetik oleh sangkar Faraday tubular, di TP setiap pasang

kabel tembaga dipelintir bersama untuk meniadakan gangguan elektromagnetik.

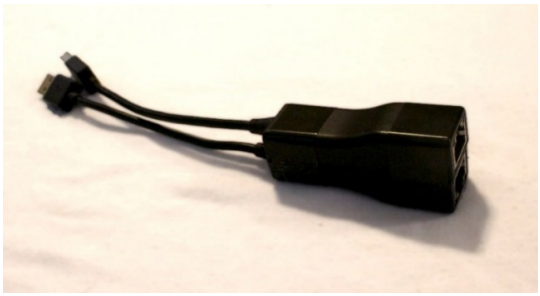


Gambar 4.23 Kabel Twisted Pair

Kabel serat optik terdiri dari helai tipis kaca atau kadang plastik yang digabungkan bersama untuk mentransmisikan sinyal. Cahaya ditransmisikan ke serat di salah satu ujungnya dan menyusuri serat optik, yang terus-menerus menempel ke dinding sampai mencapai penerima optik di ujung yang lain. Cahaya secara alami mendegradasi selama perjalanan dan bergantung pada panjang kabel serat optik, regenerator optik dapat digunakan untuk memperkuat sinyal cahaya saat transit.

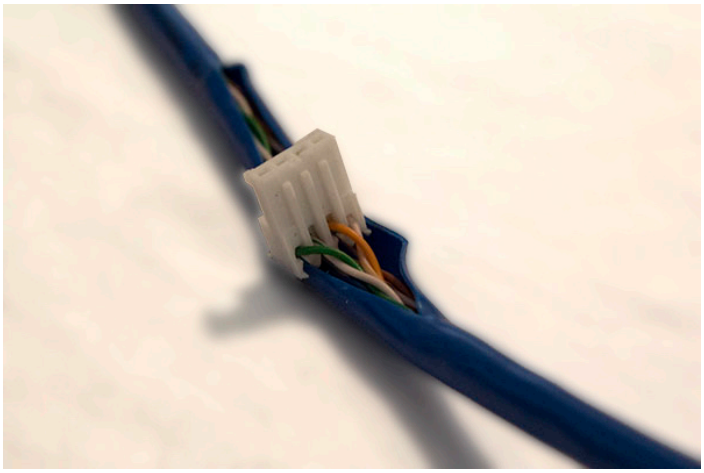
Ada berbagai alat sadap jaringan yang biasa di sebut *taps* yang tersedia untuk mencegat lalu lintas di jaringan kabel seperti:

1. *Inline network taps* adalah perangkat di *layer 1* yang dapat disisipkan sejajar antara dua perangkat jaringan yang terhubung secara fisik. *Network tap* akan melewati paket dan juga mereplikasi salinan secara fisik ke *port* terpisah atau beberapa *port*. Kerumitan jaringan biasanya memiliki empat *port*: dua terhubung *inline* untuk memudahkan lalu lintas normal, dan dua *port sniffing*, yang merefleksikan lalu lintas ke suatu arah lain. *Inline network taps* sering menyebabkan gangguan singkat karena kabel harus dipisahkan agar saat menghubungkan jaringan *tap inline*.



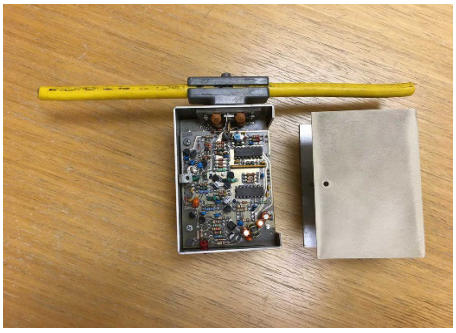
Gambar 4.24 *Inline Network Taps*

Banyak *taps* seperti terlihat pada Gambar 4.24 menggunakan perangkat keras untuk mereplikasi data, yang memungkinkan penangkapan paket yang sangat tinggi. *Taps* jaringan ini biasanya dirancang untuk tidak memerlukan daya yang besar untuk melewatkan paket secara pasif. Hal ini dimungkinkan untuk melewati lalu lintas ke *port* pemantauan tanpa menggunakan listrik seperti telihat pada Gambar 4.25, walaupun sering kali daya juga dibutuhkan untuk pemantauan.



Gambar 4.25 *Inline Network Taps* Tanpa Daya

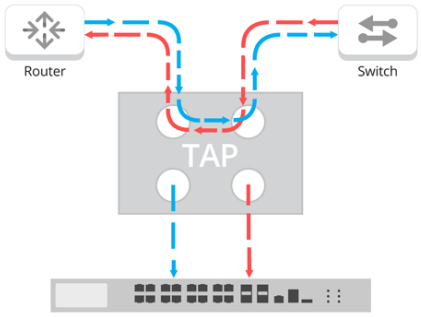
2. *Vampire taps* adalah perangkat yang dapat menembus perisai kabel tembaga untuk memberikan akses ke sinyal di dalamnya. *Vampire taps* seperti terlihat pada Gambar 4.26 tidak seperti *inline network taps*, kabel tidak perlu diputus agar *vampire taps* dapat dipasang. Namun, investigator *network forensics* harus berhati-hati karena dapat mengurangi kualias *taping*, bahkan jika dilakukan dengan benar pun dapat menurunkan kaitan pada kabel TP karena dibutuhkan karakteristik komunikasi seimbang.



Gambar 4.26 *Vampire Taps*

3. *Induction coils* atau induksi coil adalah gulungan kabel untuk menghasilkan voltase dan memancarkan berbagai sinyal elektromagnetik di luar saluran yang diinginkan. Radiasi elektromagnetik semacam itu lebih terasa pada kabel *unshielded*, seperti UTP, karena kurangnya pelindung yang diberikan selubung plastik. Sebagai konsekuensinya, secara teoretis dimungkinkan untuk mengenalkan apa yang disebut *induction coils* di samping itu kabel dapat menerjemahkan sinyal yang dipancarkan secara lateral ke dalam bentuk digital aslinya.
4. Serat optik taps seperti telihat pada Gambar 4.27 adalah memasang *taps* jaringan pada kabel serat optik, menyambungkan kabel optik dan menghubungkannya ke setiap *port taps*. Terkadang dalam

proses *taping* dapat menyebabkan gangguan jaringan. *Inline taps* optik dapat menyebabkan degradasi sinyal. Investigator *network forensics* sering menggunakan alat yang disebut *optical time-domain reflectometers* (OTDR) untuk menganalisis dan mengatasi masalah sinyal kabel serat optik.



Gambar 4.27 *Fiber Optic Taps*

OTDRs juga bisa digunakan untuk menemukan jeda di kabel. Jauh lebih sulit untuk men-*taps* serat optik daripada kabel tembaga. *Vampire taps* bisa menembus insulasi kabel tembaga dan terhubung secara fisik dengan kabel tembaga untuk mendeteksi perubahan tegangan di dalamnya.

4.7.1 Frekuensi Radio

Sejak akhir 1990-an, frekuensi radio telah menjadi media yang semakin populer untuk transmisi data paket dan konektivitas internet. *Institute of Electrical and Electronics Engineers* (IEEE) menerbitkan serangkaian standar internasional 802.11 untuk komunikasi *wireless local area network* (WLAN). Standar ini menentukan protokol untuk lalu lintas WLAN dalam rentang frekuensi 2,4, 3,7, dan 5 GHz. Istilah

Wi-Fi digunakan untuk merujuk pada jenis lalu lintas RF tertentu.

Gelombang RF bergerak melalui udara, yang pada dasarnya merupakan medium bersama. Akibatnya, lalu lintas WLAN tidak dapat dibagi secara fisik dengan cara mengalihkan lalu lintas segmen ke LAN berkabel. Karena keterbatasan media fisik, semua transmisi WLAN dapat diamati dan dicegat oleh semua *host*/perangkat dalam jangkauannya. Perangkat/*host* dapat menangkap lalu lintas RF, terlepas dari apakah mereka berpartisipasi dalam tautan. Atribut ini membuat perolehan lalu lintas WLAN pasif sangat mudah, baik bagi investigator *network forensics* maupun penyerang.

Di Indonesia sesuai dengan Peraturan Menteri Komunikasi dan Informatika Nomor 28 Tahun 2015 tentang Persyaratan Teknis Alat dan Perangkat Telekomunikasi, yang boleh beroperasi hanya pada Pita Frekuensi Radio 2,4 GHz dan/atau Pita Frekuensi Radio 5,8 GHz. Di Amerika Serikat, Komisi Komunikasi Federal telah membatasi kekuatan emisi untuk stasiun yang beroperasi pada rentang frekuensi 802.11 dan gain antena. Akibatnya, ada keterbatasan praktis pada jarak, di mana yang secara hukum dapat menangkap dan menerima data melalui jaringan 802.11.

Namun, jika tersangka sudah terlibat dalam aktivitas terlarang, tidak ada alasan untuk menganggap mereka tidak akan melangkah lebih jauh dalam kegiatan kriminal mereka tersebut. rmanno Pietrosemoli mengumumkan bahwa timnya telah berhasil mentransfer data melalui Wi-Fi menempuh jarak 238 mil.

Mengapa hal ini penting bagi investigator *network forensics*? Pertama, investigator *network forensics* harus mengingat bahwa target mungkin dapat mengakses WLAN dari jarak jauh, jauh di luar batas fisik WLAN. Kedua, investigator *network forensics* harus ingat bahwa ketika terhubung melalui tautan nirkabel, aktivitas bisa berpotensi dipantau dari jarak yang sangat jauh. Jika lalu lintas Wi-Fi dienkripsi,

biasanya ada satu tombol pra-berbagi disebut *pre-shared key* (PSK) untuk semua *host*. Dalam kasus ini, siapa pun yang telah memperoleh akses ke kunci enkripsi dapat mendengarkan semua lalu lintas yang berkaitan dengan semua *host* atau perangkat seperti halnya hub fisik. Untuk investigasi, ini sangat membantu karena staf TI lokal dapat memberikan kredensial autentikasi, yang memudahkan pemantauan. Lebih jauh lagi, ada kekurangan yang terkenal dalam *algoritma* enkripsi 802.11 umum, seperti *wired equivalent privacy* (WEP). Untuk hal-hal yang rumit, jalur akses nirkabel menggunakan berbagai standar yang berbeda untuk enkripsi dan autentikasi seperti 802.11n.

Dimungkinkan menangkap data atau *evidence* secara pasif melalui lalu lintas Wi-Fi yang terenkripsi dan mendeskripsinya secara *offline* menggunakan kunci enkripsi. Setelah investigator *network forensics* mendapatkan akses penuh ke konten lalu lintas 802.11x yang tidak dienkripsi, data ini dapat dianalisis dengan cara yang sama seperti lalu lintas jaringan lain yang tidak terenkripsi. Terlepas dari apakah lalu lintas Wi-Fi dapat diacak, investigator *network forensics* dapat memperoleh banyak informasi dengan menangkap dan menganalisis lalu lintas manajemen 802.11. Informasi ini umumnya meliputi:

- SSID
- Alamat MAC WAP
- Didukung *algoritma* enkripsi/autentikasi
- Alamat *client* MAC yang terkait
- Dalam banyak kasus, isi paket *layer 3*+ penuh

Untuk menangkap lalu lintas nirkabel, investigator *network forensics* pertama-tama harus memiliki perangkat keras yang diperlukan. Banyak adapter dan *driver* jaringan 802.11 standar tidak mendukung untuk mode monitor, yang memungkinkan pengguna untuk menangkap semua paket pada jaringan yang telah mendukung mode monitor.

Adaptor jaringan juga harus mendukung protokol 802.11 tertentu yang digunakan, misalnya kartu 802.11a/b/g tidak selalu mendukung 802.11n. Periksa model adaptor 802.11 *network* Anda dapat membaca *driver* yang sesuai untuk sistem operasi Anda.

Gambar 4.28 *USB-Wireless Mode Monitor*

Adapter jaringan 802.11 yang tersedia cukup banyak secara komersial, seperti terlihat pada Gambar 4.28, merupakan adapter yang dirancang khusus untuk menangkap paket dengan mode monitor. Adaptor ini termasuk fitur yang sangat berguna untuk investigator *network forensics*, seperti kemampuan untuk beroperasi sepenuhnya secara pasif sehingga investigator *network forensics* tidak perlu khawatir tentang transmisi data yang tidak disengaja, konektor untuk antena ekstra, dan faktor bentuk portabel seperti USB.

4.7.2 Hub dan *Switch*

Hub merupakan perangkat jaringan yang secara fisik berada pada *layer 1* yang menghubungkan semua stasiun pada subnet lokal ke satu rangkaian jaringan. Sebuah hub tidak menyimpan cukup banyak alamat untuk melacak apa yang terhubung dengannya, atau tidak berisi pengetahuan tentang perangkat apa yang terhubung ke *port* apa. Hub hanya dirancang sebagai perangkat fisik untuk melaksanakan konektivitas jaringan secara *baseband* atau bersama.

Saat hub menerima *frame*, ia mentransmisikan ulangnya ke semua *port* lainnya. Oleh karena itu, setiap perangkat yang terhubung ke hub secara fisik menerima semua lalu lintas yang ditujukan ke setiap perangkat lain yang terhubung ke hub. Jika sebuah hub ada di jaringan maka semua *host* atau *station* dapat terhubung dan menangkap semua paket yang meliwati hub tersebut. Investigator *network forensics* harus berhati-hati saat menggunakan hub sebagai alat penangkap lalu lintas seperti terlihat pada Gambar 4.29. Investigator *network forensics* yang melihat semua lalu lintas di segmen ini, di sisi lain bisa juga dilihat dan ditangkap data yang melewati jaringan oleh orang lain. Setiap bukti yang dikirim ke investigator *network forensics* melalui lalu lintas normal mungkin di jaringan lokal dengan memanfaatkan hub yang sudah terpasang di jaringan, namun memasang hub untuk tujuan pengambilan lalu lintas dapat menambah risiko baru secara tidak perlu. Intinya bagi investigator *network forensics*: jika menggunakan hub, Anda mungkin ingin memperoleh keuntungan darinya. Namun ingatlah jika menggunakan hub, apa pun yang Anda lihat, dan apa pun yang Anda kirim, dapat dilihat oleh orang lain di jaringan juga.

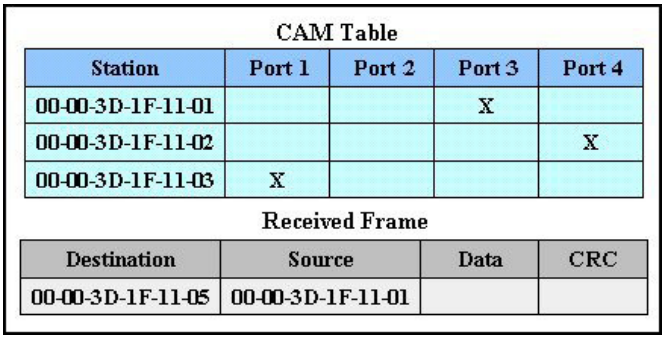
Gambar 4.29 Hub

Switch adalah perangkat *layer 2* yang paling umum seperti terlihat pada Gambar 4.30. Seperti hub, mereka juga menghubungkan beberapa stasiun bersama untuk membentuk LAN. Tidak seperti hub, switch menggunakan perangkat lunak untuk melacak stasiun mana yang terhubung ke *port* mana, di tabel CAM-nya. Ketika sebuah switch menerima sebuah paket, ia meneruskannya hanya ke stasiun tujuan. Stasiun individu tidak secara fisik saling menerima lalu lintas masing-masing. Ini berarti bahwa *port eve* pada switch adalah *collision* domainnya sendiri. Switch beroperasi pada *layer 2/data-link layer*, dan terkadang *layer 3/network layer*. Bahkan switch sederhana mempertahankan tabel CAM, yang menyimpan alamat MAC dengan *port switch* yang sesuai. Alamat MAC adalah pengenal yang diberikan ke kartu jaringan masing-masing stasiun. Tujuan tabel CAM adalah untuk memungkinkan peralihan mengisolasi lalu lintas secara *port- byport* sehingga masing-masing stasiun hanya menerima lalu lintas yang ditujukan khusus untuknya, dan bukan lalu lintas yang ditujukan untuk komputer lain.

Gambar 4.30 Switch

Cara kerja switch mengisi tabel CAM adalah dengan mendengarkan lalu lintas paket yang tiba. Saat sebuah switch menerima *frame* dari sebuah perangkat, ia melihat alamat MAC sumber dan mengingat *port* yang terkait dengan alamat MAC tersebut. Kemudian, ketika switch menerima paket yang ditujukan untuk perangkat itu, ia akan

mencari alamat MAC dan *port* yang sesuai di tabel CAM. Kemudian mengirimkan paket hanya ke *port* yang sesuai, dienkapsulasi dengan alamat Ethernet di *layer 2*.



CAM Table

Station	Port 1	Port 2	Port 3	Port 4
00-00-3D-1F-11-01			X	
00-00-3D-1F-11-02				X
00-00-3D-1F-11-03	X			

Received Frame

Destination	Source	Data	CRC
00-00-3D-1F-11-05	00-00-3D-1F-11-01		

Gambar 4.31 Cara kerja Switch Mengisi Tabel CAM

Penyidik dapat menangkap *evidence* dari lalu lintas jaringan dengan menggunakan switch. Meskipun secara *default*, switch hanya mengirim lalu lintas ke *port* tujuan menunjukkan di dalam tabel CAM seperti ditunjukkan pada Gambar 4.31, switch dengan kemampuan perangkat lunak yang memadai dapat dikonfigurasi untuk mereplikasi lalu lintas dari satu atau lebih *port* ke beberapa *port* lain *port mirroring* untuk agregasi dan analisis. Switch memiliki berbagai kemampuan *mirroring port*, tergantung modelnya. Menemukan sebuah switch yang mampu melakukan beberapa *mirroring port*, namun untuk itu kemampuan dibatasi oleh jumlah *port* yang dapat *mirrorkan*.

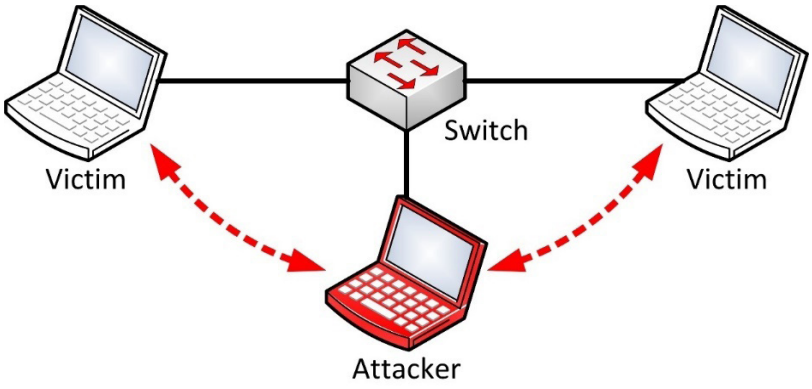
Port mirroring secara *inheren* dibatasi oleh kapasitas fisik switch itu sendiri. Sebagai contoh, katakanlah memiliki switch 100Mbps dan penyidik mencoba memantulkan *empat port*, yang masing-masing melewati rata-rata 50Mbps ke *port* SPAN tunggal. Jumlah lalu lintas dari keempat *port* menambahkan hingga 200Mbps, yang jauh di atas kapasitas dari satu *port* yang akan diterima. Hasilnya adalah

oversubscription, dan paket akan digagalkan oleh switch. Penyidik atau investigator memerlukan akses administratif ke sistem operasi switch untuk mengkonfigurasi *mirroring port*. Begitu penyidik telah melihat *por*t yang menarik, Anda dapat menghubungkan *sniffer ke port mirroring* dan menangkap semua lalu lintas.

Jika penyidik tidak memiliki akses administratif, apakah masih mungkinkan untuk mengendus lalu lintas dari sebuah switch?

Mari pelajari metode yang digunakan penyerang. Dalam kasus yang terjadi, seperti ketika administrator jaringan itu sendiri tidak dipercaya, penyidik mungkin perlu menggunakan teknik yang sama seperti penyerang. Ini bukan metode teraman. Namun, menyebabkan peralihan beroperasi di luar parameter normal, namun bisa berfungsi. Untuk mengendus lalu lintas dari sebuah switch, penyerang menggunakan salah satu dari dua metode umum. Pertama, penyerang bisa membanjiri switch dengan informasi palsu untuk tabel CAM dengan mengirimkannya banyak paket Ethernet dengan alamat MAC yang berbeda. Serangan ini disebut sebagai *MAC flooding*. Begitu tabel CAM terisi, banyak switch secara *default* akan *fail, open* dan mengirim semua lalu lintas untuk sistem yang tidak ada di tabel CAM ke setiap *port*.

Kedua, penyerang bisa melakukan serangan *ARP spoofing*, seperti terlihat pada Gambar 4.32. Biasanya, *address resolution protocol* (ARP) digunakan oleh stasiun pada LAN untuk secara dinamis memetakan alamat IP (*layer 3*) ke alamat MAC yang sesuai. Dalam serangan *spoofing* ARP, penyerang *broadcasts* paket ARP palsu yang menghubungkan alamat MAC penyerang ke alamat IP korban. Stasiun lain di LAN menambahkan informasi palsu ini ke tabel ARP mereka dan mengirim lalu lintas untuk alamat IP *router* ke alamat MAC penyerang.



Gambar 4.32 ARP *Spoofing*

4.8 Konsep Bukti Digital

Perubahan Undang-Undang Republik Indonesia Nomor 11 Tahun 2008 tentang Informasi dan Transaksi Elektronik dengan Undang-Undang Republik Indonesia Nomor 19 Tahun 2016 tentang Perubahan Atas Undang-Undang Nomor 11 Tahun 2008 tentang Informasi dan Transaksi Elektronik telah meletakan konsep dan memberikan dasar hukum mengenai kekuatan hukum tentang bukti digital sebagai syarat formil dan materil alat bukti agar dapat diterima di persidangan. Apa itu definisi bukti digital menurut undang-undang yang berlaku di Indonesia. Bukti digital adalah satu atau sekumpulan alat elektronik dan atau hasil keluaran dari alat elektronik (informasi elektronik dan/atau dokumen elektronik) yang telah memenuhi suatu persyaratan formil dan persyaratan materil secara ilmiah dan bisa dipertanggungjawabkan di pengadilan.

Dalam Pasal 5 ayat (1) yang menjelaskan tentang Perubahan Undang-Undang Republik Indonesia Nomor 11 Tahun 2008 tentang Informasi dan Transaksi Elektronik dengan Undang-Undang Republik Indonesia Nomor 19 Tahun 2016 tentang Perubahan Atas Undang-Undang Nomor 11 Tahun 2008 tentang Informasi dan Transaksi

Elektronik, menyatakan bahwa informasi elektronik dan/atau dokumen eektronik dan/atau hasil cetaknya merupakan perluasan dari alat bukti hukum yang sah sesuai dengan hukum acara yang berlaku di Indonesia. Berikut bunyi-bunyi pasal yang terkati dengan bukti digital dan jaringan komputer dari Pasal 1 ayat (1) (2) (3) (4) (5) (6) (6a) (7) dan (8) secara lengkap:

Pasal 1 ayat (1)

Informasi elektronik adalah satu atau sekumpulan data elektronik, termasuk tetapi tidak terbatas pada tulisan, suara, gambar, peta, rancangan, foto, electronic data interchange (EDI), surat elektronik (electronic mail), telegram, teleks, telecopy atau sejenisnya, huruf, tanda, angka, kode akses, simbol, atau perforasi yang telah diolah yang memiliki arti atau dapat dipahami oleh orang yang mampu memahaminya.

Pasal 1 ayat (2)

Transaksi elektronik adalah perbuatan hukum yang dilakukan dengan menggunakan komputer, jaringan komputer, dan/atau media elektronik lainnya.

Pasal 1 ayat (3)

Teknologi informasi adalah suatu teknik untuk mengumpulkan, menyiapkan, menyimpan, memproses, mengumumkan, menganalisis, dan/atau menyebarkan informasi.

Pasal 1 ayat (4)

Dokumen elektronik adalah setiap informasi elektronik yang dibuat, diteruskan, dikirimkan, diterima, atau disimpan dalam bentuk analog, digital, elektromagnetik, optikal, atau sejenisnya, yang dapat dilihat, ditampilkan, dan/atau didengar melalui komputer atau sistem elektronik, termasuk tetapi tidak terbatas pada tulisan, suara, gambar, peta, rancangan, foto atau sejenisnya, huruf, tanda, angka, kode akses, simbol atau perforasi

yang memiliki makna atau arti atau dapat dipahami oleh orang yang mampu memahaminya.

Pasal 1 ayat (5)

Sistem elektronik adalah serangkaian perangkat dan prosedur elektronik yang berfungsi mempersiapkan, mengumpulkan, mengolah, menganalisis, menyimpan, menampilkan, mengumumkan, mengirimkan, dan/atau menyebarkan informasi elektronik.

Pasal 1 ayat (6)

Penyelenggaraan sistem elektronik adalah pemanfaatan sistem elektronik oleh penyelenggara negara, orang, badan usaha, dan/atau masyarakat.

Pasal 1 ayat (6a)

Penyelenggara sistem elektronik adalah setiap orang, penyelenggara negara, badan usaha, dan masyarakat yang menyediakan, mengelola, dan/atau mengoperasikan sistem elektronik, baik secara sendiri-sendiri maupun bersama-sama kepada pengguna sistem elektronik untuk keperluan dirinya dan/atau keperluan pihak lain.

Pasal 1 ayat (7)

Jaringan sistem elektronik adalah terhubungnya dua sistem elektronik atau lebih, yang bersifat tertutup ataupun terbuka.

Pasal 1 ayat (8)

Agen elektronik adalah perangkat dari suatu sistem elektronik yang dibuat untuk melakukan suatu tindakan terhadap suatu informasi elektronik tertentu secara otomatis yang diselenggarakan oleh orang.

Sedangkan bukti digital secara global menurut *Kamus Inggris Oxford Compact* mendefinisikan bukti sebagai: bukti (kata benda).

1. Informasi atau tanda-tanda yang menunjukkan apakah suatu kepercayaan atau proposisi itu benar.

2. Informasi yang digunakan untuk menetapkan fakta dalam penyelidikan hukum atau dapat diterima sebagai kesaksian di pengadilan hukum.

Tujuan dalam banyak penyelidikan adalah mengumpulkan sekumpulan bukti yang sesuai untuk dipresentasikan dalam proses. Tujuan pertama adalah memastikan fakta-fakta tentang masalah ini dan memahami dengan benar dan benar apa yang telah terjadi.

Akibatnya, kita mendefinisikan bukti dalam arti seluas-luasnya sebagai peristiwa yang dapat diamati dan dapat direkam, atau artefak sebuah peristiwa yang dapat digunakan untuk membangun pemahaman tentang penyebab dan motif dari kejadian tersebut.

Berikut ini ada beberapa kategori bukti digital:

- *Real*
- *Best*
- *Direct*
- *Circumstantial*

4.8.1 *Real Evidence* atau Bukti Digital

Apa itu bukti "nyata", secara kasar didefinisikan sebagai objek fisik dan berwujud yang memainkan peran yang relevan dalam suatu peristiwa yang sedang diadili. Contohnya, pisau itu ditarik dari tubuh korban, ini adalah pistol yang melepaskan peluru, ini adalah salinan fisik dari kontrak yang ditandatangani oleh kedua belah pihak. Di *digital forensics* contohnya *hard drive* fisik di mana datanya telah dipulihkan dan atau semua komponen fisik komputer lainnya terlibat.

Gambar 4.33 Contoh Barang Bukti Kejahatan

Bukti nyata biasanya terdiri dari fisik peristiwa tersebut, seperti terlihat pada Gambar 4.33, dan karena itu sering kali merupakan unsur kejahatan yang paling mudah dipresentasikan dan dipahami. Manusia memahami benda-benda berwujud jauh lebih mudah, lain hal saat benda-benda tersebut tersaji dalam bentuk digital seperti data yang terdiri dari rangkaian satu dan nol seperti terlihat ilustarsi di Gambar 4.34 di bawah ini.

Gambar 4.34 Ilustrasi dari Digital *Evidence*

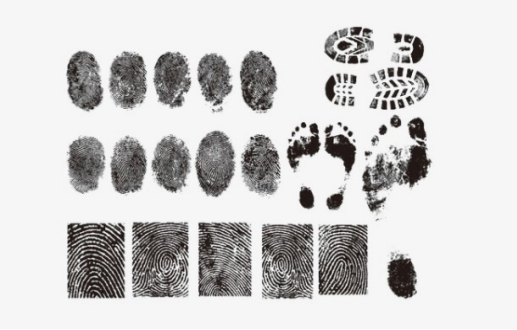
Sebagai contoh, semisal sebuah *hard drive* digunakan sebagai objek tumpul dalam serangan yang menyebabkan kematian pada korban, dari hasil oleh tempat kejadian perkara (TKP) kemungkinan bisa ditemukan bercak darah dari korban, sudut *hard drive* yang mengenai tubuh korban dan yang paling penting mungkin dapat ditemukan sidik jari dari pelaku penyerangan. Akan sangat berbeda perlakuan terhadap *hard drive* yang terlibat dalam sebuah aksi kejahatan *cyber* atau digital. Tidak akan ditemukan bukti fisik dari pelaku seperti: sidik jarik, percikan darah atau sudut *hard drive* yang dipakai untuk menyerang korban.

Kesimpulan pada *real evidence* atau buktinya adalah satu atau sekumpulan alat fisik, baik itu bersifat elektronik atau tidak. Berikut ini disebutkan beberapa contoh bukti nyata dapat mencakup:

1. Senjata pembunuhan, seperti terlihat pada Gambar 4.35 di bawah ini.

Gambar 4.35 Senjata Pembunuhan

2. Sidik jari atau tapak, ditunjukkan pada Gambar 4.36 di bawah ini.



Gambar 4.36 Sidik Jari atau Tapak (Li, n.d.)

3. Kertas kontrak yang ditandatangani.
4. Komputer, *hard drive* fisik atau perangkat USB.

4.8.2 *Best Evidence*

Best Evidence atau bukti terbaik secara kasar didefinisikan sebagai bukti terbaik yang bisa dibuat dan bisa dipresentasikan di pengadilan. Jika bukti asli tidak tersedia maka bukti bisa digunakan. Misalnya, jika kontrak yang ditandatangani asli dihancurkan, namun ada duplikat, maka duplikatnya dapat diterima. Namun, jika yang asli ada dan bisa diakui maka duplikatnya tidak perlu digunakan. Ilustrasi favorit kami tentang *peraturan bukti terbaik* berasal dari Dr. Eric Cole, seperti yang disajikan dalam kursus SANS-nya: Bayangkan sebuah helikopter dan *trailer* traktor bertabrakan di sebuah jembatan. Bukti nyata dalam kasus ini adalah helikopter, *trailer* dan reruntuhan jembatan, tetapi tidak ada harapan untuk membawa semua bukti nyata ke dalam deposisi, apalagi di depan dewan hakim atau juri. Dalam kasus seperti itu, foto-foto

pemandangan terjadi tabrakan dari rekaman terbaik yang bisa dibawa ke pengadilan.

Analis forensik, pengacara, dan hakim telah mempertanyakan apa yang merupakan bukti asli dalam kasus bukti digital. Yang asli dari sebuah tulisan atau rekaman berarti penulisan atau rekaman itu sendiri atau rekan yang dimaksudkan memiliki efek yang sama dengan orang yang mengeksekusi atau menerbitkannya. Untuk informasi yang tersimpan secara elektronik, asli berarti hasil cetakan atau keluaran lainnya dapat dibaca oleh penglihatan, jika informasi tersebut mencerminkan secara akurat. Yang asli dari sebuah foto termasuk yang negatif atau yang dicetak darinya. Sebuah duplikat sesuatu yang dihasilkan oleh proses mekanis, fotografi, kimia, elektronik, atau proses atau teknik setara lainnya yang secara akurat mereproduksi yang asli.

Dengan kata lain, sebuah cetakan dari *hard drive* komputer yang secara akurat mencerminkan data biasanya dianggap sebagai bukti asli. Dalam forensik jaringan, bit dan byte yang disajikan telah dicatat dan dapat diperlakukan dengan cara yang sama seperti foto dari sebuah peristiwa. Seolah-olah kita telah memotret perjalanan peluru melalui udara. Perbedaannya adalah penyidik forensik jaringan sering kali bisa merekonstruksi salinan keseluruhan peluru forensik dari keseluruhan foto tersebut. Contoh bukti terbaik meliputi:

1. Foto tempat kejadian perkara (TKP).
2. Salinan kontrak yang ditandatangani.
3. *File* ditemukan dari *hard drive*.
4. Sebuah *snapshot bit-for-bit* dari sebuah transaksi jaringan.

4.8.3 *Direct*

Bukti langsung adalah kesaksian yang ditawarkan oleh saksi langsung tentang tindakan atau tindakan yang dimaksud. Ada banyak cara agar peristiwa dapat diamati, ditangkap, dan dicatat di dunia nyata, dan sistem peradilan kita mencoba mengakomodasi sebagian besar fakta-fakta berikut bila ada bukti yang relevan. Tentu saja, metode tertua adalah pengamatan reportase sesama manusia. Kesaksian manusia ini diklasifikasikan sebagai *bukti langsung*, dan ini tetap merupakan bentuk bukti yang paling banyak digunakan, jika hal itu sering diperdebatkan dan tidak dapat dipercaya. Bukti langsung biasanya bisa diterima, asalkan itu relevan. Apa yang disaksikan orang lain dapat berdampak besar pada sebuah kasus yang disebut dalam konteks hukum adalah sebagai saksi.

4.8.4 *Circumstantial*

Circumstantial evidence atau bukti tidak langsung adalah bukti yang tidak secara langsung mendukung kesimpulan tertentu. Sebaliknya, bukti tidak langsung dapat dihubungkan bersama dengan bukti lain dan digunakan untuk menyimpulkan sebuah kesimpulan.

Bukti sungguhan penting untuk kasus yang melibatkan forensik jaringan karena ini adalah mekanisme utama yang digunakan untuk menghubungkan bukti elektronik dan penciptanya. Sering kali, bukti tidak langsung digunakan untuk membuat *email*, *log* obrolan, atau bukti digital lainnya. Pada gilirannya, verifikasi diperlukan untuk menetapkan keaslian, yang diperlukan agar bukti dapat diterima di pengadilan.

4.10 Tantangan Bukti Digital di *Network Forensics*

Bukti berbasis jaringan menimbulkan tantangan khusus di beberapa bidang, termasuk akuisisi, konten, penyimpanan, privasi, perampasan,

dan penerimaan. Kita akan membahas beberapa tantangan umum di bawah ini.

1. Akuisisi

   Sulit menemukan bukti spesifik di lingkungan jaringan. Jaringan berisi begitu banyak kemungkinan sumber bukti dari titik akses nirkabel ke *proxy* web ke server *log* pusat yang terkadang menunjukkan lokasi yang benar dari suatu bukti. Bahkan ketika Anda tahu di mana bukti tertentu berada, Anda mungkin mengalami kesulitan untuk mendapatkan akses karena alasan politis, tempat atau teknis.

2. Konten

   Tidak seperti *filesystem*, yang dirancang untuk menampung semua isi *file* dan metadata mereka, perangkat jaringan mungkin tidak menyimpan bukti dengan tingkat perincian yang diinginkan. Perangkat jaringan sering kali memiliki kapasitas penyimpanan yang sangat terbatas. Biasanya, hanya metadata pilihan tentang transaksi atau transfer data yang disimpan alih-alih catatan lengkap data yang dilalui jaringan.

3. Penyimpanan

   Perangkat jaringan umumnya tidak menggunakan penyimpanan sekunder atau persisten. Sebagai konsekuensinya, data yang dikandungnya mungkin sangat mudah berubah karena tidak dapat bertahan dalam pengaturan ulang perangkat.

4. Privasi

   Bergantung pada yurisdiksi, mungkin ada masalah hukum yang melibatkan privasi pribadi yang unik untuk teknik akuisisi berbasis jaringan.

4.11 Soal Latihan

1. Jelaskan secara singkat defini network forensic analysis tools (NFAT) dan contoh *software*-nya!
2. Jelaskan secara singkat defini *vulnerability assessment* dan contoh *software*-nya!
3. Jelaskan secara singkat cara kerja dari *intrusion detection system* (IDS)!
4. Sebutkan dan analisis pasal-pasal dalam UU tentang konsep bukti digital!
5. Jelaskan cara kerja penyerang dalam mengendus paket yang melewati sebuah switch!

Bab 5
Network Forensics Format Akuisisi dan Analisis

Tujuan Instruksional Umum:

1. Mahasiswa/wi mampu menjelaskan tentang format akuisisi di *network forensics*.
2. Mahasiswa/wi mampu menjelaskan tentang teknik analisis di *network forensics*.

Tujuan Instruksional Khusus:

1. Mahasiswa/wi mampu memahami tentang pentingnya akuisisi di *network forensics*.
2. Mahasiswa/wi mampu memahami berbagai format koleksi dan pelestarian bukti digital.
3. Mahasiswa/wi mampu memahami memahami format *file capture packet*, pcap.
4. Mahasiswa/wi mampu memahami akuisisi bukti di tingkat *router*.
5. Mahasiswa/wi mampu memahami latar belakang berbagai *algoritma machine learning* di *network forensics*.

Network forensics berkaitan dengan analisis jejak data dan *log* intrusi jaringan yang ditangkap oleh produk keamanan jaringan yang ada dan memberikan informasi yang berguna untuk mengkarakterisasi fitur gangguan atau perilaku buruk. Data yang dikumpulkan bertindak sebagai bukti digital untuk respons insiden dan investigasi kejahatan. *Network forensics* tidak menghalangi kejahatan jaringan. Pemantauan dan analisis data dari sistem dan jaringan akan menjadi penting bagi penegakan hukum karena beban kasus meningkat. Penjahat jaringan akan dihukum atas tindakan ilegal mereka sehingga memberikan pencegah kejahatan *online*. Kekuatan berbagai alat analisis keamanan jaringan dan forensik tersedia karena peragkat *open source* dapat diintegrasikan sehingga penyidik dapat memiliki keunggulan atas penyerang.

Tantangan sistem *network forensics* adalah mengidentifikasi informasi di jaringan yang berguna dan memilih informasi yang berpotensi untuk menjadi bukti digital yang berhubungan dengan berbagai kejahatan dunia maya. Berbagai alat keamanan dan forensik mengumpulkan data tentang atribut dan fitur protokol yang berbeda dan mencatatnya dalam format yang berbeda. Berbagai atribut yang disalahgunakan di jaringan dan lapisan *transport* protokol dapat diidentifikasi dan dianalisis. Informasi yang dikumpulkan dalam berbagai format dapat digabungkan menjadi *file* dan dianalisis untuk mendapatkan informasi bukti potensial.

Alat keamanan dan pemantauan jaringan tidak dirancang untuk menangani investigasi forensik, dan cara untuk mencapainya adalah dengan menangkap keseluruhan paket data dan menganalisisnya secara rinci. Ada dua cara untuk menangkap lalu lintas jaringan:

1. Paket dapat ditangkap dalam *file* libpcap (pcap) dengan menjalankan *packet sniffer*, seperti TCPDump.
2. NetFlow, data dikumpulkan dari *router* atau *switch*.

Tahapan ini membantu para penyidik memahami apa arti, motif serangannya, siapa yang berada di balik serangan tersebut, waktu saat diluncurkan, di mana penyerang memasuki jaringan dan bagaimana pertahanan jaringan dilanggar.

TCP/IP *protocol suite* dirancang untuk menyediakan infrastruktur komunikasi yang sederhana dan efisien. Penyerang menggunakan kerentanan dalam implementasi *stack* protokol TCP/IP dan memanfaatkannya untuk memulai serangan. Protokol penting di setiap lapisan dibahas secara singkat di bagian ini.

Protokol internet (IP), beroperasi pada *network layer* dan mengarahkan paket ke tujuannya. Paket-paket tersebut melalui serangkaian *router*, dan pada setiap *router*, *hop* ditentukan untuk paket berikutnya. Ada kemungkinan dua paket dari sumber yang sama menuju ke tujuan yang sama dapat mengambil dua jalur yang berbeda.

***Internet control message protocol* (ICMP)** memudahkan pengiriman pesan informasi satu arah ke *host*. ICMP diangkut dalam muatan paket IP dan memiliki beberapa struktur data sendiri. ICMP digunakan oleh *router* atau *host* tujuan untuk menginformasikan *host* sumber tentang kesalahan dalam pemrosesan datagram. ICMP memungkinkan *router* mengirim pesan kesalahan atau kontrol ke *router* atau *host* lain. Ini juga menyediakan komunikasi antara kedua mesin yang berkomunikasi di lapisan jaringan. Protokol ICMP digunakan untuk dua jenis operasi, melaporkan kondisi kesalahan *non-transien* dan memeriksa jaringan dengan pesan permintaan dan balasan. Pesan ICMP dikelompokkan menjadi dua kategori: pesan kesalahan ICMP dan pesan kueri ICMP.

Transmission control protocol (TCP) berjalan di atas IP dan menyediakan layanan *connection-oriented* antara sumber dan tujuan. TCP menyediakan pengiriman yang terjamin dan memastikan paket dikirim secara berurutan. Ini menggunakan berbagai mekanisme, seperti nomor urut, ucapan terima kasih, *3-way handshakes*, dan penghitung waktu.

User datagram protocol (UDP) pada dasarnya merupakan antarmuka aplikasi ke IP. Ini menyediakan sebuah mekanisme untuk satu aplikasi mengirim datagram ke yang lain. Lapisan UDP sangat tipis dan memiliki *overhead* yang rendah, namun memerlukan aplikasi untuk bertanggung jawab atas pemulihan kesalahan.

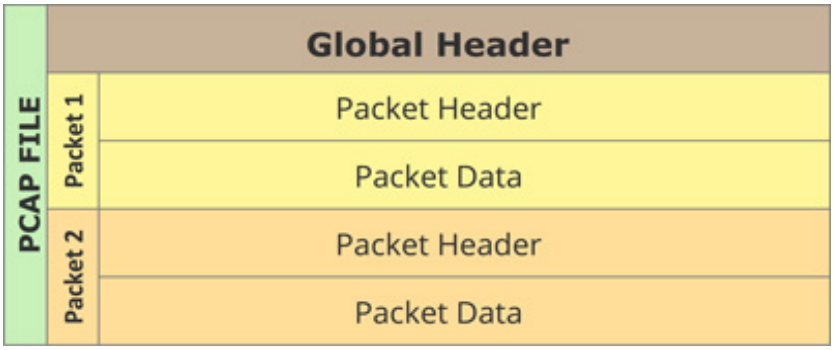
Hypertext transfer protocol (HTTP) adalah protokol ditingkat aplikasi untuk sistem informasi terdistribusi, kolaboratif, dan *hypermedia*. Ini adalah protokol generik yang dapat digunakan untuk banyak tugas di luar penggunaannya untuk *hypertext*, seperti nama server dan sistem pengelolaan objek terdistribusi, melalui perluasan metode permintaan, kode kesalahan, dan digunakan untuk mengakses data di *world wide web* (WWW). Data ditransfer antara klien dan server menggunakan pesan HTTP. Pesan HTTP dibaca dan ditafsirkan oleh server HTTP dan klien HTTP (*browser*). Format pesan permintaan dan tanggapan serupa. Pesan permintaan terdiri dari baris permintaan, tajuk, dan badan. Pesan tanggapan memiliki baris status dan bukan baris permintaan. Permintaan pesan memiliki banyak metode untuk tindakan tertentu.

5.2 Format *Capture* Paket

Network security tool dan *network monitoring tool* tidak dirancang untuk menangani penyelidikan insiden *network forensics*. Untuk mencapai hal ini, diperlukan sebuah *tool*/alat yang mampu menangkap keseluruhan paket data dan menganalisisnya secara rinci. Paket dapat ditangkap dalam *file* libpcap (pcap) dengan menjalankan *packet sniffer* seperti TCPDump. Penangkapan atau *packet capture* ini membantu para peneliti memahami apa latar belakang serangannya seperti:

1. Siapa yang berada di balik serangan tersebut.
2. Waktu saat diluncurkan.
3. Bagaimana penyerang memasuki jaringan.
4. Bagaimana merpertahankan jaringan saat terjadi serangan.

Libpcap adalah format *file* yang sangat dasar yang digunakan untuk menyimpan data jaringan yang ditangkap. Ekstensi *file* adalah pcap. *File* tersebut memiliki *header* global yang berisi beberapa informasi global yang diikuti oleh nol atau lebih catatan untuk setiap paket yang diambil.



Gambar 5.1 Libpcap File Format

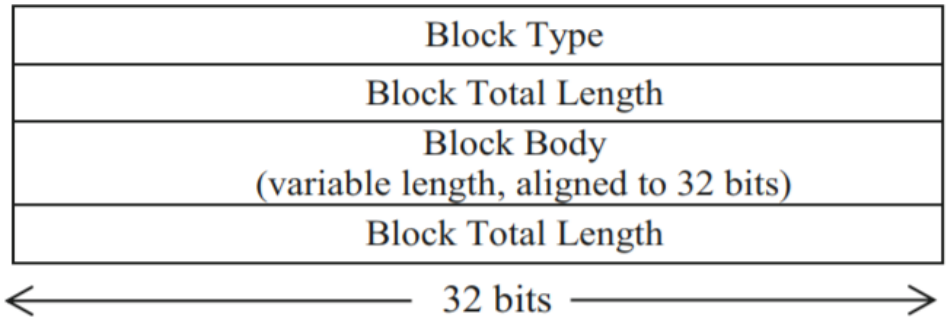
(Kurose James F.; Ross, 2013)

Paket yang diambil dalam *file* libpcap tidak berisi semua data dalam paket di jaringan, paling banyak berisi byte N pertama dari setiap paket. Nilai N disebut *snapshot length*. N akan menjadi nilai yang lebih besar dari paket terbesar yang mungkin untuk memastikan bahwa tidak ada paket dalam penangkapan yang diiris, dengan nilai tipikal 65535.

Header global ditempatkan pertama di *file* dengan menunjukkan format *file* seperti ditunjukkan pada Gambar 5.1, urutan byte, dan nomor versi. Ini menentukan waktu koreksi dalam hitungan detik antara GMT dan zona waktu setempat dan keakuratan perangko waktu dalam penangkapan. Panjang tangkapan paket N ditentukan oleh bidang *snaplen*. Jenis lapisan data *link* juga disebutkan. *Header* global diikuti oleh urutan *header* paket dan data paket. *Header* paket memiliki *field* informasi, ts_sec, yang memberi tanggal dan waktu kapan paket ini ditangkap; ts_usec, mikrodetik diimbangi dengan ts_sec saat paket ditangkap; incl_len, jumlah byte data paket yang benar-benar

ditangkap dan disimpan dalam *file*; dan bidang orig_len yang memberi panjang paket seperti pada jaringan. Data paket yang sebenarnya akan langsung mengikuti *header* paket sebagai data gumpalan incl_len byte tanpa keselarasan byte tertentu. Format libpcap sangat sederhana dan telah mendapatkan pengguna yang luas. Namun terbatas tidak memiliki resolusi waktu untuk tingkat nanodetik dan juga tidak dapat menampilkan rincian koneksi yang spesifik, informasi *interface* dan *packet drop count*.

Pcap next generation (NG) adalah format baru untuk membuang jejak paket, memiliki kemampuan untuk menggabungkan dan menambahkan data. *File capture* diatur dalam blok yang ditambahkan satu ke yang lain untuk membentuk *file*. Semua blok berbagi format umum, yang ditunjukkan pada Gambar 5.2. Jenis blok adalah nilai unik yang mengidentifikasi blok. Panjang total blok memberi ukuran total blok dalam satuan byte. Bidang ini diduplikasi pada akhirnya untuk memungkinkan navigasi ke belakang. Isi blok tertutup dalam bentuk *block body*.



Gambar 5.2 Pcapng File Format (Kurose James F.; Ross, 2013)

Dua blok wajib yang harus tampil setidaknya satu kali dalam setiap *file* adalah:

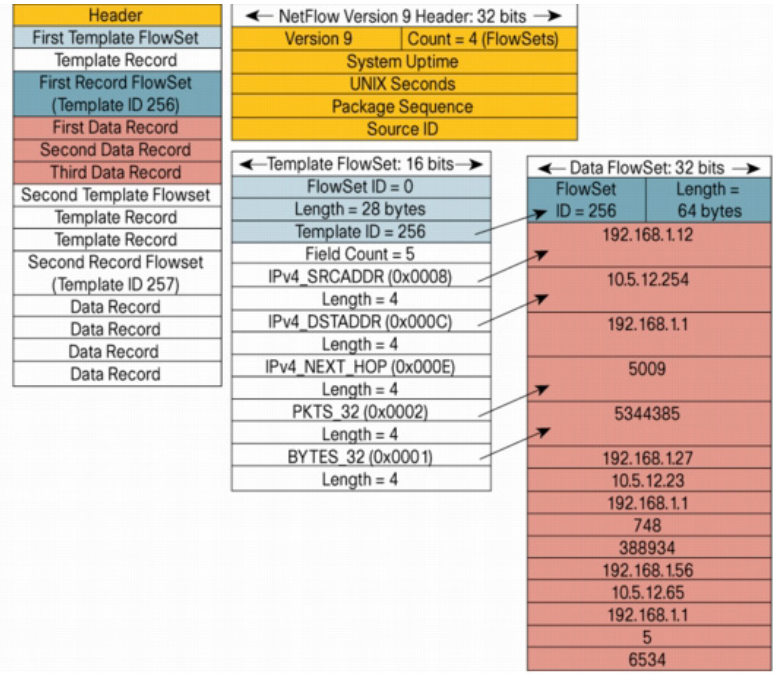
1. *Section header block* (SHB) yang mendefinisikan karakteristik terpenting dari *file capture*. Hal ini mirip dengan *header global file* libpcap.
2. *Interface description block* (IDB) yang mendefinisikan karakteristik terpenting dari antarmuka yang digunakan untuk menangkap lalu lintas. Ini berisi informasi tentang tipe *layer link* dari *interface* dan jumlah byte maksimum yang dibuang dari masing-masing paket (snaplen).

Blok opsional yang mungkin muncul dalam *file* adalah:

1. *Enhanced packet block* (EPB) yang berisi paket tertangkap tunggal atau sebagian darinya. Ini berisi informasi tentang ID antarmuka, perangko waktu, panjang yang ditangkap dan panjang paket yang sebenarnya.
2. *Simple packet block* (SPB) yang berisi satu paket yang diambil, atau sebagian darinya, hanya dengan sekumpulan informasi minimal. Ini tidak menangkap ID antarmuka dan perangko waktu.
3. *Name resolution block* (NRB) yang mendefinisikan pemetaan dari alamat numerik yang ada di *packet dump* dan mitra nama *kanonik*. Ini menghindari masalah permintaan DNS setiap kali penangkapan dibuka.
4. *Interface statistics block* (ISB) yang mendefinisikan bagaimana menyimpan beberapa data statistik, berguna untuk memahami kondisi di mana pengambilan telah dilakukan.

NetFlow NetFlow *record format* menyediakan informasi akses jaringan mengenai lalu lintas IP dalam jaringan. Digunakan untuk menghitung, audit dan juga dapat digunakan untuk deteksi intrusi,

Network forensics dan memerangi serangan DDoS. *Output* dasar NetFlow adalah *flow record*. Evolusi dari format terbaru saat buku ini di tulis pada Januari 2018 ini adalah versi 9, yang berbasis *template*.



Gambar 5.3 NetFlow *Record Format*

NetFlow terdiri dari paket ekspor seperti terlihat pada Gambar 5.3, paket dibangun oleh perangkat yang mengaktifkan layanan NetFlow dan ditujukan ke perangkat lain yang mengumpulkan dan memproses paket. Bagian pertama dari paket ekspor adalah *header* paket yang menyediakan informasi tentang versi NetFlow, jumlah *record*, *uptime* sistem, UTC detik, urutan penomoran dan ID sumber. FlowSet adalah kumpulan catatan yang mengikuti tajuk paket dalam paket ekspor. Terdiri dari dua jenis, *template* dan data. Catatan FlowSet *template* mendefinisikan format data berikutnya yang diterima dalam

paket ekspor mendatang. ID FlowSet membedakan catatan *template* dari catatan data. Catatan *template* memiliki nilai antara 0 dan 255 dan membantu dalam memproses data NetFlow tanpa harus mengetahui format data terlebih dahulu. Setiap catatan *template* dapat dibedakan dengan nomor unik yang disebut ID *template*.

Bidang panjang memberi panjang total FlowSet. *Field count* memberikan jumlah bidang dalam catatan *template* ini. *Field type* diberikan dengan nilai numerik yang mewakili jenis *field* yang merupakan vendor tertentu. Cisco memberikan nilai yang konsisten di semua platform. Data FlowSet *record* memberikan informasi tentang aliran IP yang ada pada perangkat yang menghasilkan paket ekspor. Setiap data FlowSet merujuk pada ID *template* yang dikirim sebelumnya. Data *record* memiliki FlowSet ID lebih besar dari 255. Record N, Field N menentukan kumpulan nilai *field*. Jenis dan panjangnya telah ditentukan dalam catatan *template* menggunakan *count* lapangan.

5.3 *Network Forensic Analysis*

Analisis forensik jaringan atau *network forensic analysis* adalah aktivitas yang dilakukan oleh penyidik untuk merekonstruksi aktivitas jaringan selama suatu periode. Pendekatan ini biasanya digunakan untuk menyelidiki individu yang dicurigai melakukan kejahatan dan merekonstruksi rangkaian aktivitas kejadian berbasis jaringan. *Machine learning* merupakan salah satu pendekatan populer yang digunakan untuk menganalisis kejadian jaringan yang menambahkan kekuatan pada komputer untuk mengadopsi dan bereaksi sesuai situasi berdasarkan *algoritma* ini digunakan untuk membangun model dan membuat prediksi berdasarkan pengalaman sebelumnya. *Algoritma machine learning* digunakan secara luas dalam data *mining*, klasifikasi

pola, bidang medis, dan deteksi intrusi untuk menganalisis lalu lintas jaringan. *Machine learning* diklasifikasikan ke dalam tiga kategori:

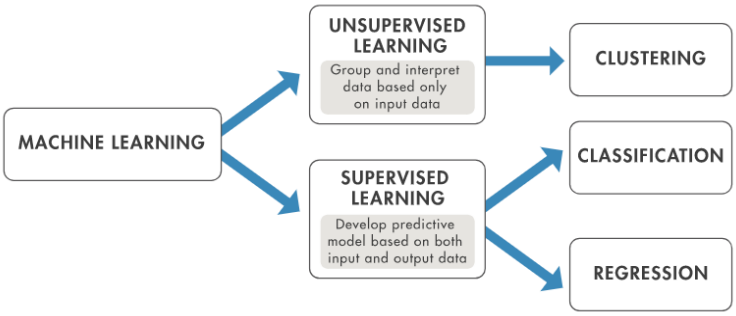
1. *Supervised learning*
2. *Unsupervised learning*
3. *Reinforcement learning*

Algoritma supervised learning machine learning adalah *algoritma* pembelajaran yang diawasi masukan datanya dengan menggunakan seperangkat aturan dan tanggapan yang diketahui data *output*-nya dan melatih model untuk menghasilkan prediksi yang masuk akal untuk respons terhadap data masukan baru. *Supervised learning* menggunakan teknik klasifikasi dan regresi untuk mengembangkan model prediktif.

Teknik klasifikasi memprediksi respons diskrit misalnya, apakah *email* itu asli atau spam, atau apakah tumor itu kanker atau tidak berbahaya. Model klasifikasi mengklasifikasikan data masukan ke dalam kategori. Aplikasi yang umum termasuk pencitraan medis, pengenalan ucapan, dan penilaian kredit. Teknik regresi memprediksi tanggapan terus-menerus misalnya, perubahan suhu atau fluktuasi permintaan daya, contoh yang umum termasuk peramalan beban listrik. *Algoritma* regresi yang umum meliputi: *decision trees*, *neural networks*, dan *adaptive neuro-fuzzy learning*.

Algoritma yang ada di *supervised learning* seperti:

1. *Naive Bayes*
2. *Decision tree*
3. *Support vector machine* SVM
4. *K-Nearest-Neighbor* KNN



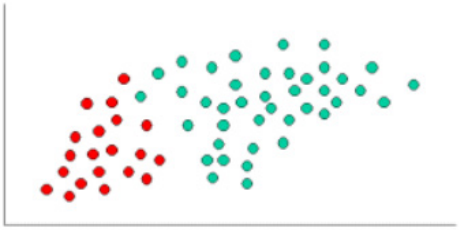
Gambar 5.4 *Machine Learning*

Pengelompokan pada *network forensic analysis* dibagi ke dalam dua kategori:

1. ***Misuse Detection***

   Misuse detection adalah salah satu pendekatan deteksi intrusi yang didasarkan pada pembuatan profil perilaku pola serangan dan kerentanan sistem yang diketahui. Ada berbagai metode untuk membuat profil intrusi sistem. *Machine learning* adalah salah satu metode untuk menganalisis pola serangan kumpulan data yang disediakan dan membuat profil umum dari lalu lintas serangan. Beberapa teknik *machine learning* yang digunakan *misuse detection* adalah:

   Metode naive bayes diusulkan oleh Thomas Bayes (1702-1761). Teknik naive bayes classifier didasarkan pada apa yang disebut teorema Bayesian dan sangat sesuai bila dimensi *input* tinggi. Meskipun sederhana, naive bayes sering kali bisa mengungguli metode klasifikasi yang lebih canggih.



Gambar 5.5 Naive Bayes

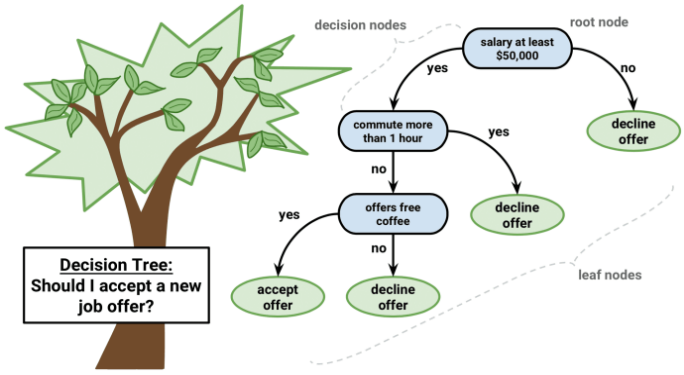
(Budiharto, 2016)

Untuk menunjukkan konsep *naïve bayes classification*, perhatikan contoh yang ditunjukkan pada Gambar 5.5 di atas. Objek dapat diklasifikasikan sebagai GREEN atau RED, tugasnya adalah mengklasifikasikan kasus baru saat mereka tiba, yaitu memutuskan label kelas mana yang menjadi milik mereka, berdasarkan objek yang saat ini ada.

Decision tree (DT) digunakan sangat luas dalam pengumpulan data dan *machine learning* untuk mengekstraksi informasi yang bermakna dari kumpulan data masukan. Membangun struktur seperti pohon di mana masing-masing cabang merepresentasikan fitur yang berbeda dan simpul daun mewakili label kelas, seperti ditunjukkan pada Gambar 5.6. *Decision Tree* mampu memecahkan keputusan yang kompleks, membuat proses menjadi kumpulan keputusan yang lebih sederhana, sehingga memberikan solusi yang sering kali lebih mudah untuk ditafsirkan. Konsep pohon keputusan membagi dan menaklukkan dengan mode *top-down*. Aliran eksekusi *decision tree* adalah sebagai berikut:

1. Ambil seluruh rangkaian *input*.
2. Temukan atribut untuk membagi *input* tersebut untuk memaksimalkan ukuran kemurnian.

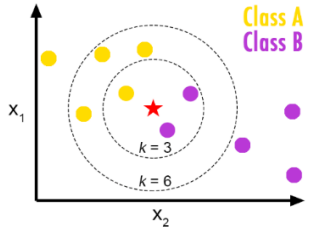
3. Bagi masukan itu berdasarkan atribut pemisahan. Ulangi langkah 1-3 untuk setiap perpecahan.
4. Pemangkasan terakhir dilakukan untuk menghindari/ menghapus *overfitting*.



Gambar 5.6 Contoh Penerapan *Decision Tree*

Langkah 2 seperti yang disebutkan di atas sangat penting dalam *algoritma decision tree*. Peneliti mendefinisikan kriteria untuk menemukan atribut pemisahan. Beberapa metrik terkenal untuk menemukan atribut yang lebih informatif dan menemukan atribut pemisahan terbaik seperti :ID3, C4.5, dan C5.0 menggunakan metrik ini.

Nearest neighbor (NN) dikenal dengan teknik "*lazy classification*", hanya mencari contoh serupa atau yang terdekat untuk mengklasifikasikan pengamatan baru ke dalam kategori yang sesuai, berdasarkan pengamatan klasifikasi atau kumpulan data pelatihan. Ukuran kesamaan dapat dihitung dengan menggunakan jarak seperti Euclidean jarak jauh. NN adalah teknik belajar yang diawasi atau *supervised learning* yang terdiri dari tahap pelatihan dan pengujian. Objek data disimpan dalam ruang n-dimensi dengan label yang sesuai pada tahap pelatihan.

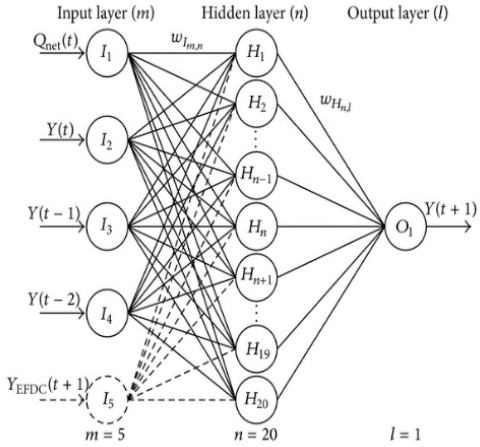


Gambar 5.7 Nearest Neighbor

Data yang tidak berlabel diberikan dalam tahap pengujian, dan *algoritma* menghitung metrik jarak, dan objek baru (lalu lintas jaringan) diberi label atau kelas tetangga terdekat dengan jarak minimum atau label atau kelas yang paling populer di tetangga terdekat (kNN) di set pelatihan seperti terlihat pada Gambar 5.7.

Banyak peneliti menunjukkan minat mereka untuk menggunakan penggolong NN karena mengklasifikasikan objek setelah menghitung semua pasangan jarak yang mungkin antara semua data pelatihan dan catatan kumpulan data. Transformasi data dari nilai atribut kontinu dalam rentang [0, 1] dihitung dengan menggunakan teknik standar yang dikenal dengan normalisasi *min-max*, di mana *minx* mewakili nilai minimum dan *maxx* adalah nilai atribut maksimum X.

Back propagation neural network **(BPNN)** adalah *algoritma* jaringan syaraf yang sangat banyak digunakan, memiliki banyak lapisan dan setiap node memiliki setidaknya satu atau lebih node yang saling berhubungan dengan beberapa fungsi aktivasi. Pada *algoritma* back propagation seperti terlihat pada Gambar 5.8, lapisan masukan akan berada di sisi paling kiri, sedangkan *output*-nya berada pada lapisan paling kanan, dan mungkin ada memiliki satu atau lebih lapisan tersembunyi atau sering disebut *hidden layer* di antara keduanya.



Gambar 5.8 Back Propagation Neural Network

(Norvig & Russell, 2010)

Pola disajikan ke lapisan masukan yang berkomunikasi dengan lapisan tersembunyi, dan pemrosesan sebenarnya terjadi melalui serangkaian koneksi tertimbang. Propagasi balik bekerja baik dalam arah ke depan dan ke belakang. Awalnya, perhitungan arah ke depan dilakukan dari *input* ke *layer output* (melalui *hidden layers*), dan setelah itu perhitungan mundur dilakukan pada arah yang berlawanan, yaitu dari *output* ke *input*.

Pada *back propagation*, simpul *output* dihitung dari sejumlah daerah yang berbeda, dianjurkan untuk menggunakan notasi yang tidak biasa untuk mewakili daerah yang berbeda, yaitu untuk setiap keluaran hanya satu simpul yang dapat memiliki nilai 1. Oleh karena itu, jumlah keluaran harus kurang dari jumlah daerah yang berbeda. Dalam *algoritma* ini, setiap kali sebuah vektor masukan dari sampel pelatihan disajikan, vektor keluaran o dibandingkan dengan nilai yang diinginkan d. Perbandingan dilakukan dengan menghitung

perbedaan kuadrat dari dua (1). Nilai Err memberitahu kita seberapa jauh kita berasal dari nilai yang diinginkan untuk *input* tertentu. Tujuan propagasi balik adalah meminimalkan jumlah Err untuk semua sampel pelatihan, sehingga jaringan berperilaku paling diinginkan (2). Kita dapat mengekspresikan Err dalam hal vektor masukan (i), vektor bobot (w), dan fungsi ambang dari persepsi. Dengan menggunakan fungsi kontinu (bukan fungsi langkah) sebagai fungsi ambang batas, kita dapat mengekspresikan gradien Err sehubungan dengan w dalam w dan i. Dengan fakta bahwa penurunan nilai w ke arah gradien menyebabkan penurunan Err yang paling cepat, kami memperbarui vektor bobot setiap kali sampel disajikan dengan menggunakan rumus persamaan berikut seperti yang diberikan di bawah ini:

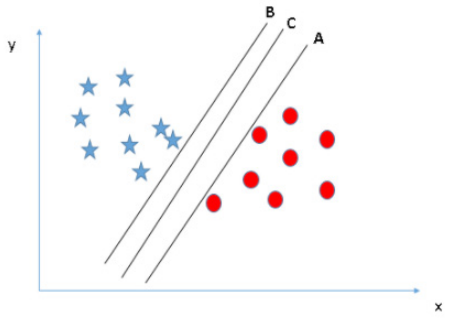
$$Err = (d - o)^2$$

$$\text{Minimize} \sum Err = (d - o)^2$$

$$w_{\text{new}} = w_{\text{old}} - n(\delta Err/\delta w)$$

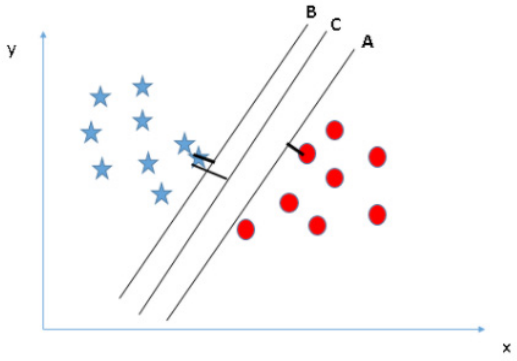
Support vector machine **(SVM)** dikembangkan oleh Corinna dkk. di AT & T Bell Labs, Holmdel, yang didasarkan pada konsep teori pembelajaran statistik dan *hiperplanes optimal*. Pengklasifikasi diskriminatif yang didefinisikan secara formal oleh sebuah penghubung yang terpisah. Dengan kata lain, diberi data pelatihan berlabel *supervised learning*, *algoritma* menghasilkan hyperplane optimal yang mengategorikan contoh baru. Dalam dua ruang dimensi *hyperplane* ini adalah garis yang membagi dua bagian di mana pada masing-masing kelas terbentang di kedua sisi. Berikut ini skenario-skenario yang akan memperjelas bagaimana cara kerja dari SVM:

1. Identifikasi *hyper-plane* kanan (skenario-1): Di sini, kita memiliki tiga bidang *hiper-pesawat* (A, B dan C). Selanjutnya, identifikasikan *hyper-plane* kanan untuk mengklasifikasikan bintang dan lingkaran seperti terlihat pada Gambar 5.9.



Gambar 5.9 *Support Vector Machine* (SVM) (Alpaydin, 2013)

2. Ditunjukkan pada Gambar 5.10, memaksimalkan jarak antara titik data terdekat (kelas satu) dan *hyper-plane*, ini akan membantu kita menentukan *hyper-plane* yang tepat. Jarak ini disebut margin.



Gambar 5.10 Skenario 2 (Alpaydin, 2013)

Di Gambar 5.10 di atas, kita melihat bahwa margin untuk *hyper-plane* C, lebih tinggi dibandingkan dengan A dan B. Oleh karena itu, kami memberi nama *hyper-plane* yang tepat sebagai C, alasan lain untuk memilih *hyper-plane* dengan margin yang lebih tinggi adalah *robustness*. Jika kita memilih *hyper-plane* yang memiliki margin rendah maka akan ada kemungkinan *miss-classification* yang tinggi.

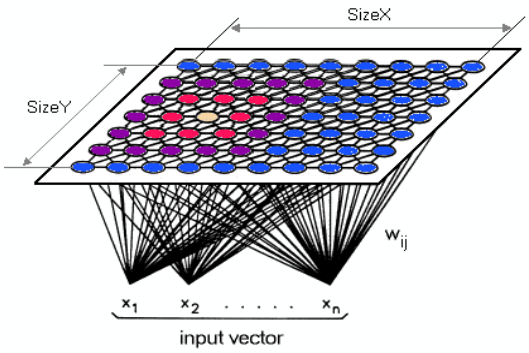
2. ***Anomaly Detection***

Anomaly detection atau deteksi anomali adalah pendekatan lain untuk melakukan analisis lalu lintas jaringan yang didasarkan pada pembuatan profil perilaku normal lalu lintas. Setiap penyimpangan dari profil mengacu pada perilaku anomali sistem. *Machine learning* adalah salah satu teknik untuk membuat profil umum perilaku normal pengguna.

Self-organizing map (SOM) dikembangkan oleh Teuvo Kohonen, memberikan teknik visualisasi data yang membantu untuk memahami data dimensi tinggi dengan mengurangi dimensi data ke peta, seperti ditunjukkan pada Gambar 5.11. SOM juga merupakan konsep pengelompokan dengan mengelompokkan data serupa secara bersamaan. Oleh karena itu dapat dikatakan bahwa SOM mengurangi dimensi data dan menampilkan kemiripan antardata.

Dengan SOM seperti ditunjukkan pada Gambar 5.11, begitu data dimasukkan ke dalam sistem, *artificial neurons*/neuron buatan dilatih dengan memberikan informasi tentang *input*. Vektor berat unit yang terdekat dengan objek saat ini menjadi unit pemenang atau aktif. Selama tahap pelatihan, nilai untuk variabel *input* secara bertahap disesuaikan dalam upaya melestarikan hubungan lingkungan yang ada di dalam kumpulan data masukan. Karena

semakin mendekati objek *input*, bobot unit pemenang disesuaikan dan juga tetangganya.



Gambar 5.11 *Self-Organizing Map*

(Budiharto, 2016)

Mendapatkan unit pencocokan terbaik dilakukan dengan menjalankan semua **vektor wright** dan menghitung jarak dari setiap bobot ke vektor sampel. Bobot dengan jarak terpendek adalah pemenangnya. Ada banyak cara untuk menentukan jarak, metode yang paling umum digunakan adalah *jarak euclidean* dan/atau *consine distance*. Singkatnya, pembelajaran terjadi dalam beberapa langkah dan banyak iterasi:

1. Setiap bobot node diinisialisasi.
2. Sebuah vektor dipilih secara acak dari kumpulan data pelatihan.
3. Setiap simpul diperiksa untuk menghitung bobot mana yang paling mirip dengan vektor masukan. Simpul pemenang dikenal sebagai *best matching unit* (BMU).
4. Kemudian lingkungan BMU dihitung. Jumlah tetangga menurun seiring berjalannya waktu.
5. Bobot kemenangan dihargai dengan menjadi lebih seperti vektor sampel. Tetangga juga menjadi lebih seperti vektor

sampel. Semakin dekat sebuah simpul menuju BMU, semakin banyak bobotnya berubah dan semakin jauh tetangganya berasal dari BMU, semakin sedikit yang dipelajari.

Algoritma Apriori dikembangkan oleh Agrawal dan Srikant pada tahun 1994. Aturan asosiasi adalah salah satu *algoritma* data *mining* berbasis aturan yang menjadi populer di analisis *supermarket* yang bertujun untuk menemukan keteraturan dalam perilaku belanja pelanggan. Tujuan utama *algoritma* ini adalah untuk mengetahui pola dan korelasi yang sering terjadi di antara berbagai *item* yang ada dalam *database*. Dua faktor kunci *algoritma apriori* adalah dukungan dan kepercayaan diri, yang digunakan untuk menemukan hubungan yang paling penting. Untuk aturan XY, dukungan mengacu pada seberapa sering *item* (X dan Y) muncul di *database*. Keyakinan mengacu pada seberapa sering *item* Y muncul dalam transaksi yang mengandung X.

Apriori adalah salah satu pendekatan penambangan data yang paling populer untuk menemukan *frequent itemset* dari kumpulan data transaksi dan menurunkan aturan asosiasi. Menemukan *frequent itemset* (*itemset* dengan frekuensi yang lebih besar dari atau sama dengan minimum *support* yang ditentukan pengguna) tidak sepele karena ledakan kombinatorialnya. Setelah *frequent itemset* diperoleh, sangat mudah untuk menghasilkan aturan asosiasi dengan keyakinan lebih besar dari atau sama dengan keyakinan minimum yang ditentukan pengguna. Menggunakan *algoritma apriori*, yang merupakan aturan asosiasi klasik dalam sistem deteksi intrusi berbasis web dan menerapkan basis aturan yang dihasilkan oleh *algoritma apriori* untuk mengidentifikasi berbagai serangan dan meningkatkan keseluruhan kinerja sistem deteksi. *Algoritma apriori* dalam deteksi intrusi dinilai cukup banyak oleh orang akhir-akhir ini. *Algoritma apriori* yang ditingkatkan meningkatkan waktu

eksekusi sangat banyak, bila datanya kecil. Metode tradisional sering membutuhkan banyak sumber daya sistem dan dengan demikian diperlukan banyak waktu. *Algoritma* yang ditingkatkan menghitung dukungan tanpa melintasi *database*. Namun, kompleksitas *algoritma* meningkat dengan *dataset* yang lebih besar, menghabiskan banyak memori dan sumber daya prosesor.

K-Means Clustering adalah *algoritma* analisis pengelompokan yang menggabungkan objek data berdasarkan nilai atribut mereka ke dalam *cluster* K. Objek dengan nilai fitur serupa dikelompokkan ke dalam *cluster* yang sama. K adalah bilangan bulat positif, yang diberikan sebelumnya, menentukan jumlah *cluster*. Berikut ini adalah langkah-langkah *algoritma clustering* Kmeans:

1. Tetapkan jumlah kluster K.
2. Secara acak membagi semua objek data ke dalam *cluster* K dan menginisialisasi *cluster clusterer*, menghitung *cluster*, dan memverifikasi bahwa semua pusat *cluster*/*centroid* tidak berbeda satu sama lain.
3. Ulangi semua objek dan hitung jarak antarpusat dan tempatkan semua kelompok. Kemudian alokasikan setiap data objek ke *cluster* dengan *center* terdekat.
4. Hitung kembali *centroid* dari semua kelompok yang berubah.
5. Ulangi langkah 3 sampai *centroid*/*center* tidak berubah.

Pengelompokan K-means digunakan untuk menghasilkan partisi kumpulan data secara otomatis. Ini dimulai dengan memilih *centroid C cluster* awal dan kemudian berulang kali memolesnya sebagai berikut:

- Setiap contoh data xk dialokasikan ke pusat *cluster* terdekatnya.
- Rata-rata komponen komponen *cluster* dikalkulasi ulang yang menjadi pusat *cluster*/*centroid vi.*

Proses *clustering* berakhir saat tidak ada modifikasi lebih lanjut dalam penugasan objek data ke *cluster*. *Clustering* adalah proses iteratif yang tujuannya adalah untuk meminimalkan fungsi objektif.

Penelitian yang dilakukan oleh Münz dkk., mempresentasikan pendekatan pendeteksian anomali baru berdasarkan *algoritma* pengelompokan K-means. Data mentah berisi data arus lalu lintas yang diekspor oleh *router* dan pemantau jaringan. Untuk interval waktu yang telah ditentukan dan nomor *port service specific* transformasi catatan arus dilakukan ke kumpulan data dengan sejumlah kecil fitur. Proses di atas dilakukan untuk mengidentifikasi interval waktu yang menunjukkan perilaku lalu lintas anomali. Pendekatan mereka mencakup tiga langkah pemrosesan:

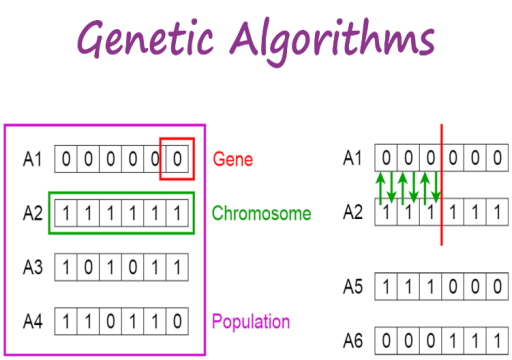
1. Langkah pertama adalah mentransformasikan data pelatihan yang berisi catatan arus lalu lintas normal dan berbahaya ke kumpulan data fitur.
2. Langkah kedua adalah menerapkan pengelompokan K-means dan membuat partisi data set yang berbeda untuk lalu lintas normal dan anomali.
3. Dan akhirnya, perhitungan jarak yang sederhana dari pusat *cluster* yang digunakan untuk mendeteksi anomali secara cepat dalam pemantauan data baru. *Clustering* mungkin mengikuti asumsi bahwa *instance* data normal membangun *cluster* yang luas dan padat, sementara anomali atau data berbahaya membuat *cluster* yang sangat kecil atau khas.

Penelitian juga dilakukan oleh Muda dkk., mempresentasikan pendekatan pembelajaran hibrida untuk deteksi anomali. Menurut mereka, pendekatan deteksi anomali mampu memprediksi serangan dengan akurasi tinggi dan tingkat deteksi tinggi. Namun tingkat *false alarm* dengan metode deteksi anomali sama tingginya. Untuk mencapai akurasi dan tingkat deteksi yang tinggi dan menurunkan

tingkat *false alarm*, mereka mengajukan kombinasi dua teknik pembelajaran.

1. Untuk tahap pertama dalam pendekatan pembelajaran hibrida yang diusulkan, mereka menggunakan pengelompokan K-means sebagai komponen pra-klasifikasi dan mengelompokkan contoh data serupa berdasarkan perilaku mereka.
2. Selanjutnya, dengan menggunakan pengklasifikasi naïve Bayes, mereka mengklasifikasikan *cluster* yang dihasilkan ke dalam kelas serangan sebagai tugas klasifikasi akhir. Mereka menemukan bahwa data yang telah salah klasifikasi selama tahap awal mungkin diklasifikasikan dengan benar pada tahap klasifikasi berikutnya.

Genetic Algorithm. John Holland awalnya menemukan *algoritma genetika* pada tahun 1960-an. *Genetic algorithm* atau *algoritma genetika* adalah pencarian heuristik yang terinspirasi oleh teori evolusi alam Charles Darwin. *Algoritma* ini mencerminkan proses seleksi alam di mana individu terkuat dipilih untuk menghasilkan keturunan generasi berikutnya seperti terlihat pada Gambar 5.12.



Gambar 5.12 *Genetic Algorithm* (Norvig & Russell, 2010)

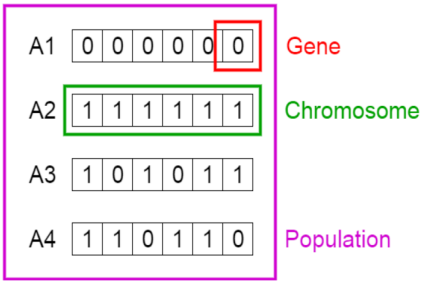
Proses seleksi alam dimulai dengan pemilihan individu terkuat dari suatu populasi. Mereka menghasilkan keturunan yang mewarisi ciri khas orang tua dan akan ditambahkan ke generasi penerus. Jika orang tua memiliki kebugaran yang lebih baik, keturunan mereka akan lebih baik daripada orang tua dan memiliki kesempatan lebih baik untuk bertahan hidup. Proses ini terus berlanjut dan pada akhirnya, generasi dengan individu terkuat akan ditemukan.

Gagasan ini bisa diterapkan untuk masalah pencarian yang mempertimbangkan satu set solusi untuk sebuah masalah dan memilih yang terbaik dari mereka.

Lima fase dipertimbangkan dalam *algoritma genetika*:

1. Populasi awal

   Proses populasi awal dimulai dengan seperangkat individu yang disebut populasi. Setiap individu adalah solusi untuk masalah yang ingin Anda selesaikan. Individu ditandai dengan seperangkat parameter (variabel) yang dikenal sebagai gen. Gen bergabung menjadi *string* untuk membentuk kromosom (solusinya). Dalam *algoritma genetika* seperti yang terlihat pada Gambar 5.13, kumpulan gen individu diwakili menggunakan *string*, dalam bentuk alfabet. Biasanya, nilai biner digunakan (string 1s dan 0s). Kita mengatakan bahwa kita mengkodekan gen dalam kromosom.



Gambar 5.13 Populasi, Kromosom, dan Gen (Norvig & Russell, 2010)

2. Fungsi kebugaran

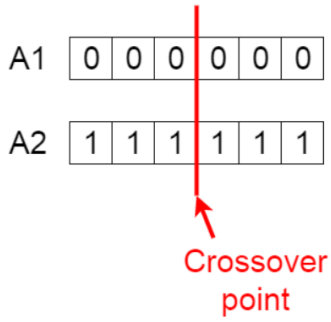
Fungsi kebugaran atau fungsi *fitness* menentukan seberapa bugar seseorang, maksudnya adalah kemampuan seseorang untuk bersaing dengan individu lain. Fungsi kebugaran ini akan memberi skor kebugaran bagi setiap individu. Probabilitas bahwa individu akan dipilih untuk reproduksi didasarkan pada skor kebugarannya.

3. Pilihan

Tahap pilihan gagasan seleksi adalah memilih individu terkuat dan membiarkan mereka melewati gen mereka ke generasi berikutnya. Dua pasang individu (orang tua) dipilih berdasarkan skor kebugaran mereka. Individu dengan kebugaran tinggi memiliki lebih banyak kesempatan untuk dipilih untuk reproduksi.

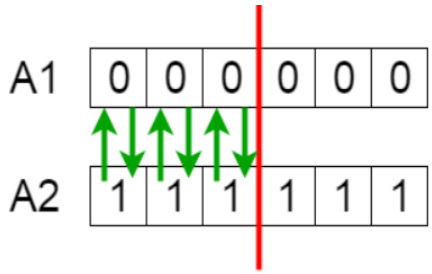
4. *Crossover*

Crossover adalah fase yang paling signifikan dalam *algoritma genetika*. Untuk setiap pasangan orang tua untuk dikawinkan, titik *crossover* dipilih secara acak dari dalam gen. Misalnya, pertimbangkan titik *crossover* menjadi tiga seperti Gambar 5.14 di bawah ini.



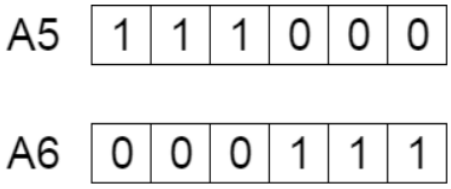
Gambar 5.14 Titik Silang (Norvig & Russell, 2010)

Keturunan diciptakan dengan menukar gen orang tua di antara mereka sampai titik *crossover* tercapai seperti terlihat pada Gambar 5.15 di bawah ini.



Gambar 5.15 Saling Menukar Gen di Antara Orang Tua (Norvig & Russell, 2010)

Terlihat di Gambar 5.16 di bawah ini, keturunan baru ditambahkan ke populasi

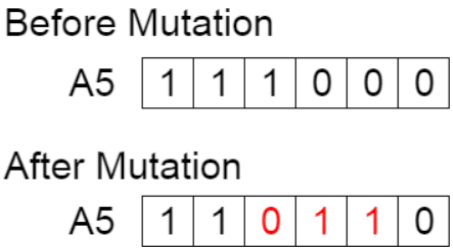


Gambar 5.16 Keturunan Baru

(Norvig & Russell, 2010)

5. Mutasi

Pada keturunan baru terbentuk, beberapa gen mereka dapat dikenai mutasi dengan probabilitas acak rendah. Ini menyiratkan bahwa beberapa bit dalam *string* bit dapat dibalik, seperti terlihat pada Gambar 5.17 di bawah ini.



Gambar 5. 17 Mutasi: Sebelum dan Sesudah (Norvig & Russell, 2010)

Mutasi terjadi untuk mempertahankan keragaman di dalam populasi dan mencegah konvergensi dini. Penghentian *algoritma* berakhir jika populasi berkumpul (tidak menghasilkan keturunan yang berbeda secara signifikan dari generasi sebelumnya). Kemudian dikatakan bahwa *algoritma genetika* telah menyediakan satu set solusi untuk masalah kita. Penerapkan *algoritma genetika* untuk deteksi intrusi, aturan sederhana adalah aturan lalu lintas jaringan yang berfungsi sebagai jalur pemisahan antara koneksi jaringan normal dan koneksi jahat. Aturan ini disimpan dalam basis aturan dalam bentuk berikut: *if condition then act*.

Dalam pernyataan di atas, bidang kondisi mengacu pada perbandingan kecocokan antara koneksi jaringan saat ini dan aturan dalam sistem deteksi intrusi yang berkemungkinan adanya gangguan, seperti di alamat IP sumber dan tujuan dan durasi koneksi, nomor *port*, protokol (TCP/IP, UDP protokol jaringan) yang digunakan, dan lain-lain. Tindakan operasi yang dilakukan oleh kebijakan keamanan suatu sistem, seperti menghentikan koneksi, mengirim pesan peringatan ke administrator sistem, dan mencatat pesan itu ke *file* audit sistem. Aturan dapat didefinisikan sebagai:

if the connection contains following information: source IP addr 124.10.5.28;dest IP addr:130.16.216.55; dest port number: 21; connection time: 15.1 seconds then terminate the connection.

Aturan di atas menjelaskan sebagai berikut: alamat IP 124.10.5.28 terdeteksi sebagai salah satu alamat IP daftar hitam oleh IDS. Jika sebuah jaringan memiliki permintaan koneksi untuk alamat IP sumber 124.10.5.28, alamat IP tujuan 130.16.216.55, nomor *port* tujuan 21, dan waktu koneksi 15.1s, kemudian hentikan hubungan itu. Setiap permintaan sambungan untuk alamat IP jahat harus ditolak. Tujuan utama penerapan GA adalah membuat peraturan seperti itu yang hanya cocok dengan koneksi jahat.

5.4 Kesimpulan

Analisis forensik jaringan melibatkan penerapan *machine learning* dan data *mining* untuk menguji dan mempelajari pola serangan. *Algoritma supervised*, *semi-supervised* dan *reinfforce learning* tidak dibahas. *Misuse detection* dan *anomaly detection* adalah dua kelas serangan deteksi yang luas. Teknik populer seperti Naïve Bayes, *decision tree*, tetangga terdekat, SVM dan lainnya dibahas. Apriori, SOM, K-Means dan mekanisme deteksi anomali lainnya juga diperkenalkan. Tantangannya adalah untuk mengetahui kombinasi *algoritma* yang tepat untuk menganalisis jenis serangan tertentu. Fase ini terdiri dari jantung forensik jaringan karena sekitar tiga perempat dari waktu digunakan untuk tahap analisis.

5.5 Soal latihan

1) Jelaskan tantangan akuisisi dalam *network forensics* dan dua cara akuisisi bukti digital di lalu lintas jaringan!
2) Jelaskan aspek-aspek pentingnya akuisisi di *network forensics*!
3) Sebutkan dan jelaskan format koleksi bukti digital yang diakuisisi dari jaringan dan metode pelestarian bukti digital
4) Jelaskan cara bagaimana akusisi barang bukti digital tingkat *router*!

Bab 6
Wireshark

Tujuan Instruksional Umum:

1. Mahasiswa/wi mampu menjelaskan secara umum tentang Wireshark.

Tujuan Instruksional Khusus:

1. Mahasiswa/wi mampu memahami tentang *system requirements* Wireshark di sistem operasi Windows, Linux dan MacOS.
2. Mahasiswa/wi mampu memahami sejarah singkat Wireshark.
3. Mahasiswa/wi mampu memahami proses instalasi Wireshark di OS Windows, Linux, dan MacOS.
4. Mahasiswa/wi mampu menangkap dan menyaring paket dengan Wireshark.

6.1 Pengantar

Apa itu Wireshark?

Wireshark adalah alat untuk analisis paket jaringan. Sebuah penganalisis paket jaringan akan mencoba untuk menangkap

paket jaringan dan mencoba untuk menampilkan data paket tersebut sedetail mungkin. Anda bisa memikirkan penganalisis paket jaringan sebagai alat ukur yang digunakan untuk memeriksa apa yang terjadi di dalam kabel jaringan, seperti voltmeter yang digunakan oleh teknisi listrik untuk memeriksa apa yang terjadi di dalam kabel listrik, namun pada tingkat yang lebih tinggi. Di masa lalu, alat semacam itu sangat mahal, eksklusif, atau keduanya. Namun, dengan kemunculan Wireshark, semua itu telah berubah. Wireshark adalah salah satu analisa paket dengan sumber terbuka terbaik yang ada saat ini.

Berikut adalah beberapa contoh penggunaan Wireshark:

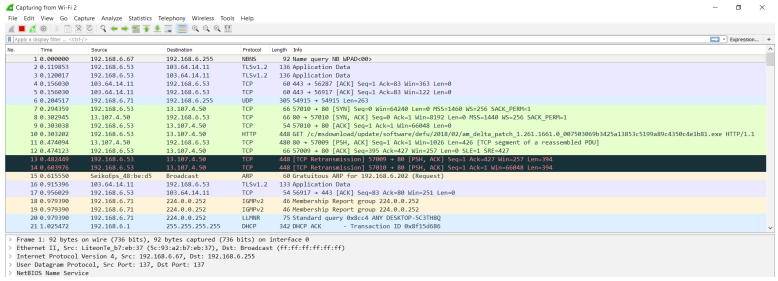
- Administrator jaringan menggunakannya untuk memecahkan masalah jaringan.
- Administrator keamanan jaringan menggunakannya untuk memeriksa masalah keamanan.
- Insinyur QA menggunakannya untuk memverifikasi aplikasi jaringan.
- Pengembang menggunakannya untuk *debug* implementasi protokol.
- *User* menggunakannya untuk mempelajari internal protokol jaringan.

Berikut adalah beberapa dari sekian banyak fitur yang disediakan Wireshark:

- Tersedia untuk UNIX dan Windows.
- Meng-*capture*/menangkap paket data langsung dari antarmuka jaringan.
- Dapat membuka *file* yang berisi data paket yang diambil dengan tcpdump/WinDump, Wireshark, dan sejumlah program *capture* paket lainnya.

- Impor paket dari *file* teks yang berisi *hex dumps* dari paket data.
- Tampilan paket dengan informasi protokol yang sangat rinci.
- Bisa meng-ekspor beberapa atau semua paket dalam sejumlah format *file capture*.
- Filter paket di banyak kriteria.
- Mencari paket dengan banyak kriteria.
- Mewarnai tampilan paket berdasarkan filter.

Wireshark dapat menangkap lalu lintas dari berbagai jenis media jaringan LAN, nirkabel dan USB juga, tergantung pada sistem operasi yang Anda gunakan seperti terlihat pada Gambar 6.1.



Gambar 6.1 Wireshark *Captures Packets*

Wireshark adalah proyek perangkat lunak sumber terbuka, dan dirilis di bawah GNU *general public license* (GPL). Anda dapat dengan bebas menggunakan Wireshark pada sejumlah komputer yang Anda sukai, tanpa perlu khawatir dengan lisensi atau biaya atau semacamnya. Selain itu, semua kode sumber tersedia secara bebas di bawah GPL. Karena itu, sangat mudah bagi orang untuk menambahkan protokol baru ke Wireshark, baik sebagai *plugin*, atau dibangun ke dalam sumbernya.

Berikut adalah beberapa hal yang tidak disediakan oleh Wireshark:

- Wireshark bukanlah *intrusion detection system* (IDS). Wireshark tidak akan memperingatkan Anda ketika seseorang melakukan hal-hal aneh di jaringan Anda sehingga dia tidak boleh melakukannya.

Namun, jika hal-hal aneh terjadi, Wireshark bisa membantu Anda mengetahui apa yang sebenarnya sedang terjadi.

- Wireshark tidak akan memanipulasi hal-hal di jaringan, hanya akan mengukur sesuatu dari jaringan. Wireshark tidak mengirim paket di jaringan atau melakukan hal-hal aktif lainnya.

6.2 *System Requirements*

Jumlah sumber daya yang dibutuhkan oleh Wireshark bergantung pada lingkungan Anda dan ukuran *file* penangkapan yang Anda akan analisis. Nilai yang bagus untuk *file capture* adalah berukuran kecil hingga menengah tidak lebih dari beberapa ratus MB. *File* pengambilan yang lebih besar akan membutuhkan lebih banyak memori dan ruang *disk*. Bekerja dengan jaringan yang sibuk bisa dengan mudah menghasilkan *file capture* yang besar. Menangkap di jaringan gigabit atau bahkan jaringan 100megabit bisa menghasilkan ratusan megabyte data *capture* dalam waktu singkat serta membutuhkan prosesor yang cepat, banyak memori dan ruang *disk* yang akan terpakai. Dampaknya jika Wireshark kehabisan sumber daya seperti memori maka ia akan macet. Berikut ini *system requirements* yang dibutuhkan oleh beberapa sistem operasi yang popular di kalangan umum:

1. Microsoft Windows

   Wireshark versi Windows mendukung versi Windows termasuk:10, 8, 7, Vista, Server 2016, Server 2012 R2, Server 2012, Server 2008 R2, dan Server 2008. Berikut ini *system requirement* untuk perangkat keras di Windows:

   - Prosesor AMD64 / x86-64 atau 32-bit x86 64-bit modern.
   - RAM yang tersedia 400 MB. *File* tangkapan yang lebih besar membutuhkan lebih banyak RAM.

- 300 MB ruang *disk* yang tersedia. *File* pengambilan membutuhkan ruang *disk* tambahan.
- Resolusi 1024×768 (1280x104 atau lebih tinggi) dengan kedalaman warna paling sedikit 16-bit.
- Kartu jaringan LAN dan *Wireless* yang mendukung untuk *capture packet*.

Wireshark 1.12 adalah rilis terakhir yang mendukung Windows Server 2003. Wireshark 1.10 adalah cabang terakhir yang secara resmi mendukung Windows XP.

2. UNIX/Linux

Wireshark berjalan pada sebagian besar platform UNIX seperti macOS dan Linux. Persyaratan sistem harus sebanding dengan nilai Windows yang tercantum di atas. Paket biner tersedia untuk kebanyakan distribusi Unics dan Linux termasuk platform berikut:

- macos Apple
- Debian GNU/Linux
- FreeBSD
- Gentoo Linux
- HP-UX
- Mandriva Linux
- NetBSD
- OpenPKG
- Red Hat Enterprise/Fedora Linux
- Sun Solaris/i386
- Sun Solaris/SPARC
- Canonical Ubuntu

Jika paket biner tidak tersedia untuk platform Anda, Anda dapat men-*download* sumbernya dan

salinan terbaru program dari situs Wireshark di https://www.wireshark.org/download.html. Halaman *download* secara otomatis akan men-*download* yang sesuai untuk platform. Versi Wireshark yang baru biasanya tersedia setiap bulan atau dua bulan.

6.3 Sejarah Singkat Wireshark

Pada akhir 1997, Gerald Combs membutuhkan alat untuk melacak masalah jaringan dan ingin belajar lebih banyak tentang jaringan sehingga dia mulai menulis Ethereal (nama asli proyek Wireshark) sebagai cara untuk menyelesaikan kedua masalah tersebut. Ethereal awalnya dirilis setelah jeda dalam pembangunan pada bulan Juli 1998 sebagai versi 0.2.0. Dalam beberapa hari *patch*, laporan dan *bug* mulai berdatangan dan Ethereal dalam perjalanan menuju kesuksesan. Tidak lama kemudian Gilbert Ramirez melihat potensinya dan menyumbangkan seorang pendidik.

Pada bulan Oktober 1998 Guy Harris mencari sesuatu yang lebih baik daripada tcpview sehingga dia mulai menerapkan *patch* dan menyumbang *dissectors* kepada Ethereal. Pada tahun 2006 proyek tersebut pindah rumah dan kembali muncul dengan nama baru: Wireshark. Pada tahun 2008, setelah sepuluh tahun pembangunan, Wireshark akhirnya sampai di versi 1.0. Rilisan ini pertama kali dianggap selesai, dengan fitur implementasi minimal. Rilisnya bertepatan dengan Wireshark Developer and User Conference pertama, yang disebut Sharkfest.

Pada tahun 2015 Wireshark 2.0 dirilis, yang menampilkan *user interface* baru.

Pengembangan dan pemeliharaan Wireshark awalnya dikembangkan oleh Gerald Combs. Pengembangan dan pemeliharaan Wireshark yang sedang berjalan ditangani oleh tim Wireshark, sekelompok individu yang memperbaiki *bug* dan menyediakan fungsionalitas baru. Ada juga

sejumlah besar orang yang telah menyumbangkan protokol *dissector* kepada Wireshark, dan diharapkan hal ini akan berlanjut. Anda dapat menemukan daftar orang-orang yang telah menyumbang kode ke Wireshark dengan mencentang kotak dialog Wireshark atau di halaman penulis di situs Wireshark.

Wireshark adalah proyek perangkat lunak sumber terbuka, dan dirilis di bawah GNU *general public license* (GPL) versi 2. Semua kode sumber tersedia secara bebas di bawah GPL. Anda dipersilakan untuk memodifikasi Wireshark agar sesuai dengan kebutuhan Anda sendiri, dan akan sangat dihargai jika Anda menyumbangkan perbaikan kembali ke tim Wireshark.

Anda mendapatkan tiga keuntungan dengan menyumbangkan perbaikan kembali ke masyarakat:

1. Orang lain yang menganggap kontribusi Anda berguna akan menghargai mereka, dan Anda akan tahu bahwa Anda telah membantu orang dengan cara yang sama seperti yang dilakukan oleh para pengembang Wireshark
2. Para pengembang Wireshark dapat memperbaiki perubahan Anda lebih jauh lagi karena selalu ada ruang untuk perbaikan. Atau mereka mungkin menerapkan beberapa hal lanjutan di atas kode Anda, yang bisa berguna bagi diri Anda juga.
3. Pemelihara dan pengembang Wireshark akan mempertahankan kode Anda juga, memperbaikinya saat perubahan API atau perubahan lainnya dilakukan, dan umumnya menjaganya agar sesuai dengan apa yang terjadi dengan Wireshark.

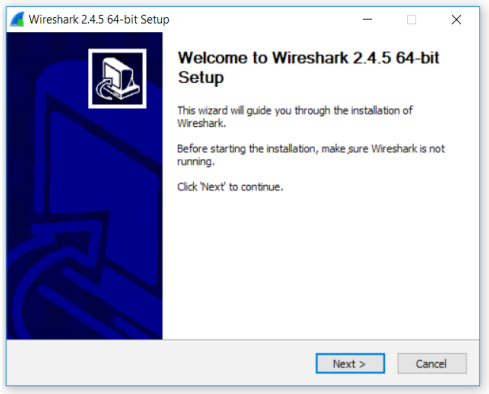
Kode sumber Wireshark dan perangkat biner untuk beberapa platform semuanya tersedia di halaman *download* situs Wireshark: https://www.wireshark.org/download.html.

6.4 *Building and Installing* Wireshark

Untuk menggunakan Wireshark Anda harus menginstalnya terlebih dahulu. Jika Anda menjalankan macOS atau Windows, Anda dapat men-*download* rilis resmi di https://www.wireshark.org/download.html dan menginstalnya seperti terlihat pada Gambar 6.2. Jika menjalankan sistem operasi lain seperti Linux atau FreeBSD yang Anda lakukan adalah menginstal dari sumber distribusi Linux yang menawarkan paket Wireshark tapi biasanya mengeluarkan versi ketinggalan zaman.

Berikut ini adalah langkah umum yang akan Anda gunakan:

1. *Download* paket yang sesuai untuk kebutuhan.
2. Kompilasi sumber menjadi biner jika dibutuhkan.



Gambar 6.2 Instalasi Wireshark di Windows

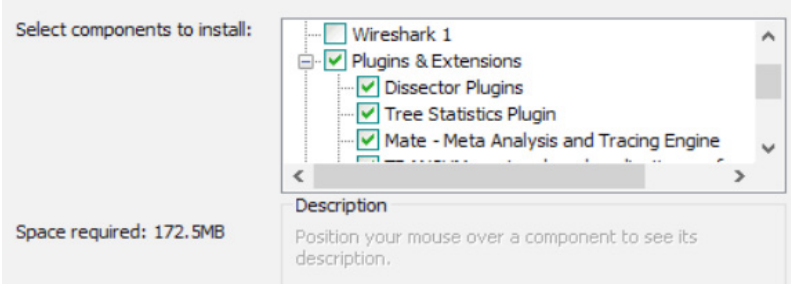
6.4.1 Menginstal Wireshark di Windows

Cukup *download installer* Wireshark dari https://www.wireshark.org/download.html dan jalankan. Anda dapat memilih untuk menginstal beberapa komponen opsional dan memilih lokasi paket yang terinstal. Pengaturan *default* direkomendasikan untuk sebagian besar pengguna.

Komponen instalasi

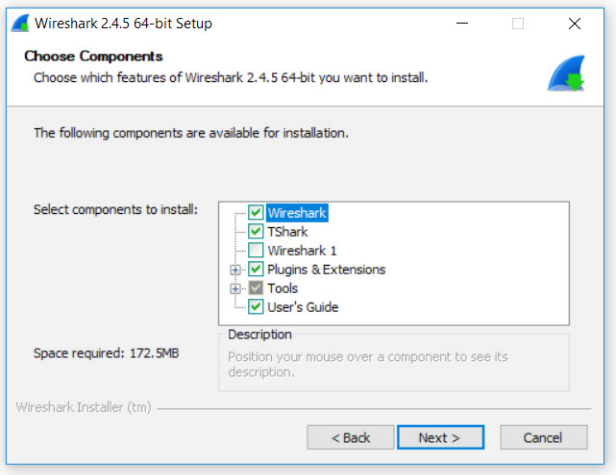
Pada halaman *Choose Components* seperti ditunjukkan pada Gambar 6.4 dari *installer* Anda dapat memilih sebagai berikut ini:

- Wireshark: penganalisis protokol jaringan.
- TShark: sebuah penganalisis protokol jaringan berbasis *command-line*.
- Wireshark 1 Legacy: Antarmuka pengguna lama (GTK +) jika Anda memerlukannya.
- Plugins & Extensions, ditunjukan pada Gambar 6.3.



Gambar 6.3 *Plugins & Extensions*

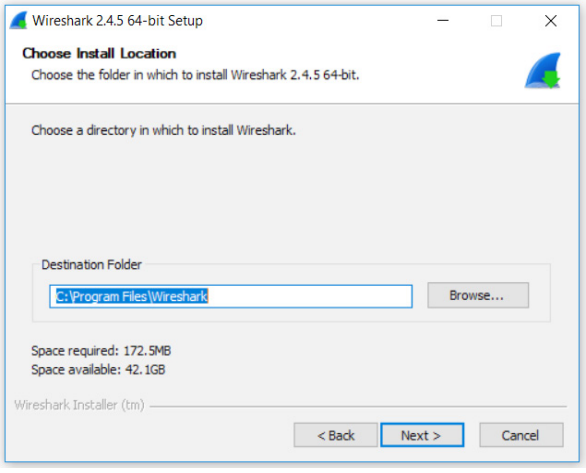
- Tool: Alat bantu tambahan yang berbasis *command line* untuk *file capture*.
- *User's guide* atau panduan pengguna: Tombol bantuan akan meminta koneksi internet untuk menampilkan halaman bantuan jika panduan pengguna tidak diinstal secara lokal.



Gambar 6.4 *Choose Components*

Secara default Wireshark menginstal ke %ProgramFiles%\ di Windows 32-bit dan

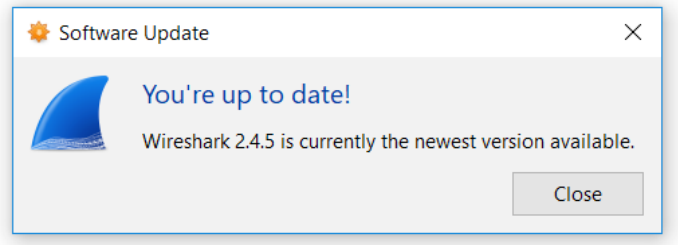
%ProgramFiles64%\ Wireshark pada Windows 64-bit. Dan ke C:\ Program Files\Wireshark pada kebanyakan sistem seperti terlihat pada Gambar 6.5.



Gambar 6.5 *Destination Folder Instalasi Wireshark di Windows*

Installer Wireshark berisi *installer* WinPcap terbaru. Jika tidak menginstal WinPcap, Anda tidak akan dapat menangkap lalu lintas jaringan langsung. Namun tetap bisa membuka *file* pengambilan yang tersimpan. Saat proses instalasi pertama kali, secara *default* versi terbaru WinPcap akan diinstal. Jika Anda tidak ingin melakukan ini atau jika Anda ingin menginstal ulang WinPcap Anda dapat memeriksa kotak *install* WinPcap jika diperlukan.

Update paket Wireshark di Windows secara *default* akan memeriksa versi baru dan memberi tahu Anda bila ada *update* tersedia, seperti ditunjukkan pada Gambar 6.6. Jika Anda memiliki preferensi cek untuk pembaruan dinonaktifkan atau jika Anda menjalankan Wireshark di lingkungan yang terisolasi Anda harus berlangganan ke milis wireshark-announce. Versi baru Wireshark biasanya dilepaskan setiap empat sampai enam minggu. Memperbarui Wireshark dilakukan dengan cara yang sama seperti menginstalnya. Cukup *download* dan jalankan *installer exe*. *Reboot* biasanya tidak diperlukan dan semua pengaturan pribadi Anda tetap tidak berubah.



Gambar 6.6 *Wireshark Status Update*

6.4.2 Instalasi Wireshark di macOS

Paket macOS resmi didistribusikan sebagai *disk images* (.dmg) yang berisi *installer* aplikasi. Untuk menginstal Wireshark cukup buka *disk image* dan jalankan *installer*. Paket *installer* termasuk Wireshark, utilitas

baris perintah terkait, dan *daemon* peluncuran yang menyesu aikan hak akses pada *startup* sistem. Untuk lebih jelasnya silakan dan baca *read me first*.

Berikut ini langkah umum untuk instalasi Wireshark di *under* UNIX atau Linux:

1. Keluarkan *file* dari *tar* terkompresi. Jika Anda menggunakan Linux atau versi UNIX Anda menggunakan *tar* GNU Anda dapat menggunakan perintah berikut :

   ```
   $ tar xaf wireshark-2.4.5.tar.xz
   ```

   Dalam kasus lain Anda harus menggunakan perintah berikut:

   ```
   $ xz -d wireshark-2.4.5.tar.xz
   $ tar xf wireshark-2.4.5.tar
   ```

2. Ubah direktori ke direktori sumber Wireshark.

   ```
   $ cd wireshark-2.4.5
   ```

3. Konfigurasi *file* sumber dengan benar untuk versi UNIX. Anda bisa melakukan ini dengan perintah berikut:

   ```
   $ ./configure
   ```

4. Mem-*build file* sumbernya.

   ```
   $ ./make
   ```

5. Instal perangkat lunak di tujuan akhirnya.

   ```
   $ ./make install
   ```

Instalasi binari di bawah UNIX Secara umum menginstal biner di bawah versi UNIX Anda akan lebih spesifik untuk metode instalasi yang digunakan dengan versi UNIX Anda. Sebagai contoh, di bawah AIX, Anda akan menggunakan *smit* untuk menginstal paket binaan Wireshark, sementara di bawah Tru64 UNIX (dahulu Digital UNIX) Anda akan menggunakan *setld*.

- Instalasi di bawah Debian, Ubuntu, dan derivatif Debian lainnya.

```
$ aptitude install Wireshark
```

Atau

```
$ dpkg -i wireshark-common_2.0.5.0-1_i386.
deb wireshark_wireshark-2.0.5.0-1_i386.deb
```

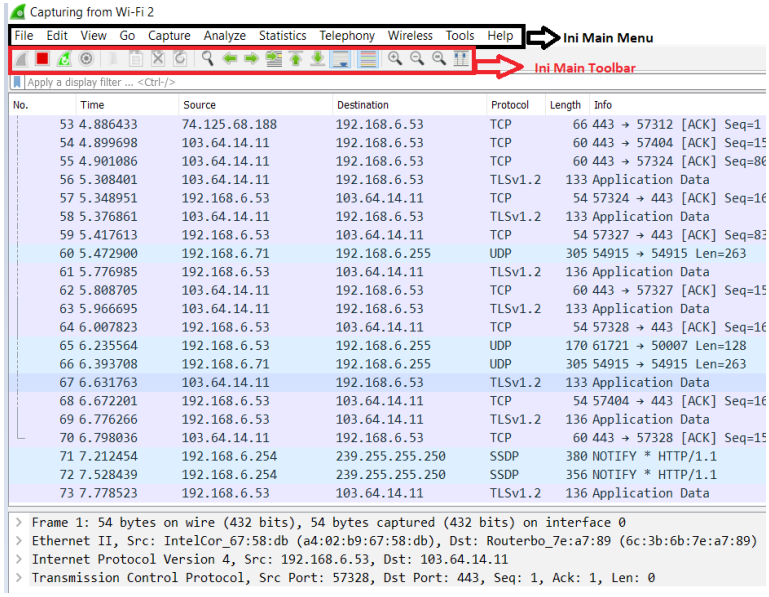
Instalasi dari paket di bawah FreeBSD.

- Gunakan perintah berikut untuk menginstal Wireshark di bawah FreeBSD:

```
$ pkg_add -r wireshark
```

6.5 *Start* Wireshark

Main window atau jendela utama Wireshark yang ditunjukkan pada Gambar 6.6, terdiri dari *Main Menu* dan *Main Toolbars*.



Gambar 6.7 *Main Windows dari Wireshark*

Berikut ini penjelasan dari bagian-bagian yang umum pada *Main Menu*:

- *File*.

  Menu ini berisi *item* untuk membuka dan menggabungkan *file* pengambilan, menyimpan, mencetak, atau mengambil *file* ekspor secara keseluruhan atau sebagian, dan untuk keluar dari aplikasi Wireshark.

- Edit

  Menu ini berisi *item* untuk menemukan paket, referensi waktu atau menandai satu atau beberapa paket, menangani profil konfigurasi, dan menetapkan preferensi Anda (*cut*, *copy*, dan *paste* tidak diterapkan saat ini).

- *View*

  Menu ini mengontrol tampilan data yang diambil, termasuk mewarnai paket, memperbesar *font*, menampilkan paket di jendela terpisah, memperluas dan meruntuhkan pepohonan dalam detail paket.

- *Go*

  Menu ini berisi *item* untuk menuju ke paket tertentu.

- *Capture*

  Dengan menu ini Anda dapat memulai dan menghentikan pengambilan dan untuk mengedit filter pengambilan.

- Analisis

  Menu ini berisi *item* untuk memanipulasi filter tampilan, mengaktifkan atau menonaktifkan pembedahan protokol, mengkonfigurasi *decode* yang ditentukan pengguna dan mengikuti aliran TCP.

► *Statistics*

Menu ini berisi *item* untuk menampilkan berbagai jendela statistik, termasuk ringkasan paket yang telah ditangkap, menampilkan statistik hierarki protokol dan masih banyak lagi.

► *Telephony*

Menu ini berisi *item* untuk menampilkan berbagai jendela statistik terkait telepon, termasuk analisis media, diagram alir, statistik hierarki protokol tampilan dan banyak lagi.

► *Wireless*

Item dalam menu ini menampilkan statistik nirkabel Bluetooth dan IEEE 802.11.

► *Tools*

Menu ini berisi berbagai alat yang tersedia di Wireshark, seperti membuat Aturan ACL *firewall*.

► *Help*

Menu ini berisi *item* untuk membantu pengguna, misal akses ke beberapa bantuan dasar, halaman manual dari berbagai alat baris perintah, akses *online* ke beberapa halaman web, dan biasanya tentang dialog.

6.6 Soal Latihan

1. Jelaskan secara singkat apa itu Wireshark!
2. Sebutkan *system requirements* Wireshark di sistem operasi Windows, Linux dan MacOS!
3. Jelaskan sejarah singkat Wireshark dari awal terbentuk sampai perjalanan sampai saat ini!
4. Jelaskan cara kerja bagaiman cara menangkap dan menyaring paket dengan Wireshark!

Bab 7
Case Study

Tujuan Instruksional Umum:

1. Mahasiswa/wi mampu melakukan analsis *network forensics*.

Tujuan Instruksional Khusus:

1. Mahasiswa/wi mampu memahami motede deteksi dalam pendeteksian *malware*.
2. Mahasiswa/wi mampu melakukan penerapan akuisisi barang bukti.
3. Mahasiswa/wi mampu menganalisis barang bukti.
4. Mahasiwa/wi mampu menyimpulkan dan mempresentasikan hasil dari analisis barang buktinya.

7.1 *Case Study*: Deteksi dan Analisis *Ransomware*

Sebuah rumah sakit di Los Angeles pada tahun 2016 terjadi infiltrasi jaringan dengan menonaktifkan jaringan dan komputer dengan menggunakan serangan *ransomware*. Penjahat *cyber* menuntut uang tebusan sebesar $17.000 untuk memulihkan jaringan dan komputer yang penuh dengan informasi penting dan konfidensial pasien. *Ransomware*

adalah salah satu jenis *malware* yang membatasi akses terhadap informasi dengan mengenkripsi *file* dan folder dengan kunci yang tidak mungkin dibuka dan selanjutnya penjahat *cyber* akan meminta uang tebusan untuk membuka akses ke *file* dan folder.

Ransomware menjadi populer di kalangan penjahat *cyber* untuk menghasilkan uang dengan cara yang mudah. *Ransomware* memiliki dampak kerusakan dan kegelisahan terhadap bisnis yang ditandai dengan meningkatnya statistik jumlah serangan *ransomware* rata-rata 100-300 persen pada 2016, dengan jumlah laporan insiden meningkat hingga 4000 persen. Pada tahun 2016 dan diperkirakan pada tahun 2017 ada tiga *ransomware*, yaitu TeslaCrypt, Locky, dan CERBER yang telah menguasai dunia *ransomware*.

Kini para pembuat *malware* membuat *ransomware* lebih canggih, lebih efektif, dan menggunakan antiforensik untuk menghindari deteksi dan analisis.

Metode deteksi *ransomware* umumnya terbagi dalam tiga pendekatan:

1. *Static feature-based*
2. *Host-based*
3. *Behavior-based network*

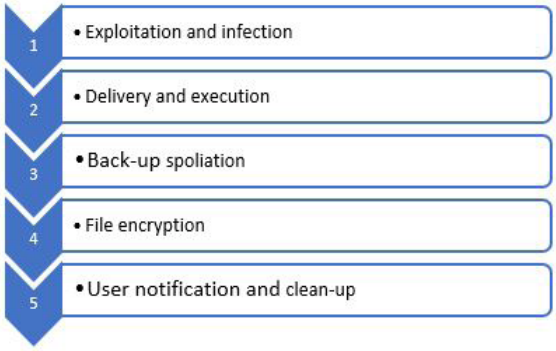
Static feature-based atau fitur berbasis statis banyak digunakan oleh perangkat lunak antivirus dan mudah dihindari oleh penyerang, seperti penyerang yang menggunakan teknik pengemasan atau perubahan struktural kode *malware* mereka. Metode *host-based* atau analisis dinamis di mana *malware* artefak dieksekusi di lingkungan VM (mesin virtual) yang juga memiliki keterbatasan karena *malware* saat ini dapat mendeteksi lingkungan VM atau komputer *host* dan juga kurang mampu mendeteksi sampel *malware* baru, dan cenderung menghasilkan peringatan palsu atau menghasilkan kesalahan klasifikasi. Metode

network forensics based behavior adalah pendekatan yang memiliki kemampuan untuk mengidentifikasi pola lalu lintas yang tidak normal selama pengoperasian jaringan.

Cerber ransomware bisa menginfeksi melalui beberapa metode yang berbeda dengan dampak yang lebih merusak dan lebih mahal. Skema umum distribusi, penyebaran dan infeksi *ransomware* melalui jaringan berbasis seperti men-*download file*, *phishing e-mail*, *download drive-by* atau situs web yang disusupi dan lain-lain.

Ransomware adalah jenis *malware* yang membatasi akses ke informasi penting perorangan atau perusahaan dengan cara mengenkripsi *file* dan akan meminta pembayaran uang tebusan dengan imbalan kunci dekripsi untuk memulihkan *file* terenkripsi. Embrio *ransomware* yang disebut PC Cyborg dimulai pada tahun 1989 oleh Dr. Joseph Popp. Setelah terinfeksi, PC Cyborg akan menyembunyikan semua *folder file* dan mengenkripsi *file* di drive C:\. Sebuah pesan naskah meminta uang tebusan sebesar $189 yang ditujukan ke PC Cyborg Corporation.

Ransomware serangan pertama menggunakan kriptografi kunci publik untuk menggabungkan kombinasi virus dan kuda *Trojan* yang disebut *cryptovirus* atau serangan *cryptovirological.* Lima fase serangan *ransomware* ditunjukkan pada Gambar 7. 1:



Gambar 7.1 Lima Fase Serangan *Ransomware*

Penjelasan dari fase lima serangan yang disebutkan di atas:

1. *Exploitation and infection*

   File ransomware perlu dieksekusi di komputer. Proses penyebaran dan infeksi sering dilakukan melalui *email phishing* atau memanfaatkan celah keamanan pada aplikasi perangkat lunak, misalnya Adobe Flash dan Internet Explorer.

2. *Exploitation and infection*

   Setelah proses *exploitation and infection*, *ransomware executable* akan dikirim kembali ke sistem korban. Setelah dijalankan, mekanisme proses ini bisa memakan waktu beberapa detik, tergantung dari latensi jaringan. *Ransomware* adalah penyebaran jaringan yang paling sering dieksekusi melalui enkripsi yang kuat dan ditempatkan di folder % APPDATA% atau % TEMP% di profil pengguna.

3. *Back-up spoliation*

   Segera setelah proses pengiriman dan eksekusi, *ransomware* akan mencari *file* dan folder cadangan dan menghapus semua *file* untuk menghindari korban yang akan mengembalikan *file* dan folder yang telah dienkripsi. Dalam sistem Windows, alat vssadmin menghapus salinan bayangan volume sistem, seperti *cryptolocker ransomware* dan *locky* akan menjalankan perintah untuk menghapus semua salinan bayangan sistem.

4. *File encryption*

   Setelah *file*, folder dan *copy* salinan bayangan benar-benar dihapus, *malware* akan melakukan pertukaran kunci aman dengan *server command and control* (C2), membangun kunci enkripsi yang hanya akan digunakan pada sistem lokal. *Ransomware* akan mengidentifikasi secara unik setiap sistem lokal untuk membedakan kunci enkripsi yang kuat di antara mereka menggunakan *algoritma*

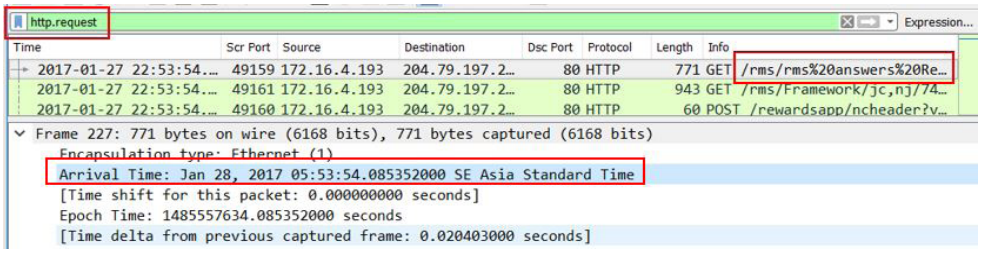
AES 256 proses enkripsi dapat berlangsung dari beberapa menit sampai jam tergantung pada latensi jaringan, jumlah, dan ukuran dokumen dan jumlah perangkat yang terhubung.

5. *User notification and clean-up*

Notifikasi pengguna dan pembersihan: dalam permintaan pemerasan fase ini dan instruksi pembayaran dipresentasikan kepada korban. Instruksi permintaan pemerasan dan disimpan ke *hard drive*, terkadang *file* instruksi di folder yang sama dengan *file* terenkripsi sebagai contoh CryptoWall versi 3 dengan nama file HELP_DECRYPT.

Pcap *dataset* silakan Anda *download* di blog saya: https://adekurniawanrusdy.wordpress.com/

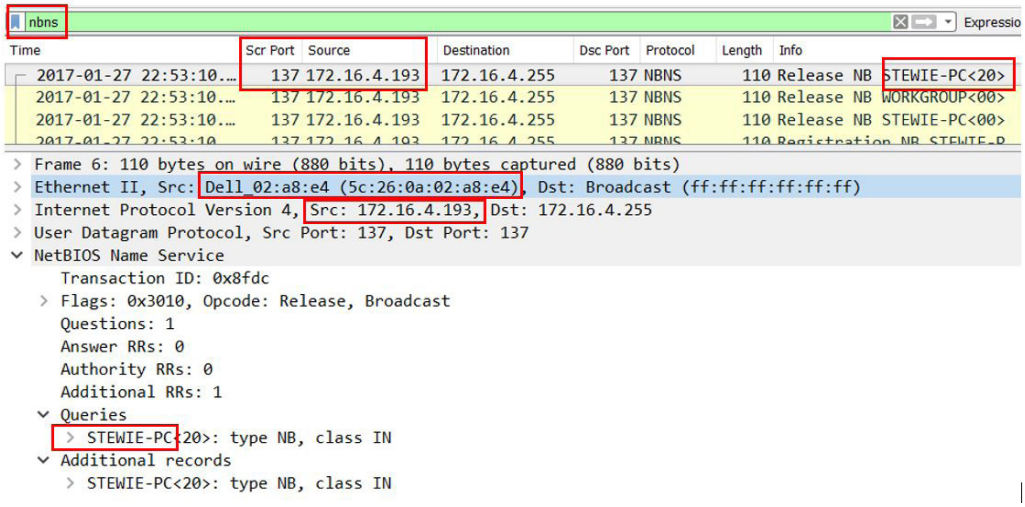
Timestamp dalam peran forensik digital sangat penting karena mengandung informasi yang berkaitan dengan pertunjukan dalam kondisi kapan saat terjadi suatu tindakan atau peristiwa. Langkah pertama yang harus dilakukan adalah menentukan kapan pertama kali komputer *host* terinfeksi, dengan menggunakan Wireshark filter http. request, seperti ditunjukkan pada Gambar 7.2 yang menunjukkan pertama kali komputer yang terinfeksi pada waktunya 2017-01-27 22 : 53: 54 UTC atau 28 Januari 2017 05:53:54 SE Asia Waktu Standar.



Gambar 7.2 Tanggal dan Waktu Infeksi

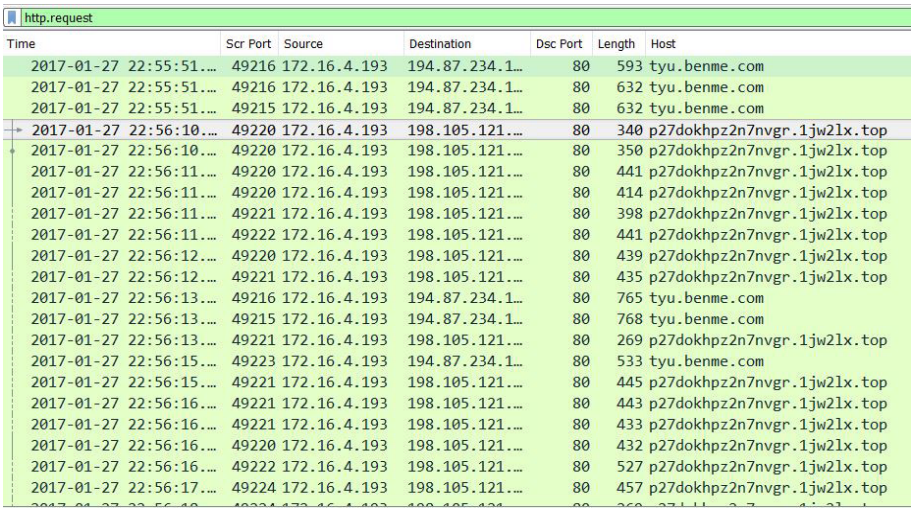
Setelah mengetahui tanggal dan waktu infeksi tahap selanjutnya adalah mendeteksi dan menganalisis IP dan nama *host* komputer yang

telah terinfeksi. IP detection, MAC Address Hostname dan NetBIOS analisis dilakukan dengan menggunakan filter nbns. NetBIOS adalah aplikasi yang memungkinkan komputer berkomunikasi dengan komputer di *local area network* (LAN). Analisis alamat IP dan MAC yang korbannya pertama kali terinfeksi ditunjukkan pada Gambar 7.3. Pengguna IP *Host Computers* yang terinfeksi adalah 172.16.4.193 dengan MAC Address 5c: 26: OA: 02: a8: e4 kartu jaringan dari vendor perangkat keras Dell dan dengan nama *Host* Stewie PC.



Gambar 7.3 Analisis Lalu Lintas NBNS di Wireshark

IP, MAC *Address* dan *hostname* yang sudah kita ketahui, tahap selanjutnya menentukan *malware* yang menginfeksi nama *host* PC Stewie . Setelah analisis mendalam dari beberapa paket yang ditunjukkan pada Gambar 7.4, lalu lintas ke domain.top yang biasa digunakan penulis/pembuat *malware* digunakan dalam melakukan kegiatan kriminal. Daftar domain yang umumnya digunakan adalah domain lclebb6kvohlkcml.onion [.] link lclebb6kvohlkcml.onion [.] nu bmacyzmea723xyaz.onion [.] link bmacyzmea723xyaz.onion [.] nu nejdtkok7oz5kjoc.onion [.] link nejdtkok7oz5kjoc.onion [.] nu.

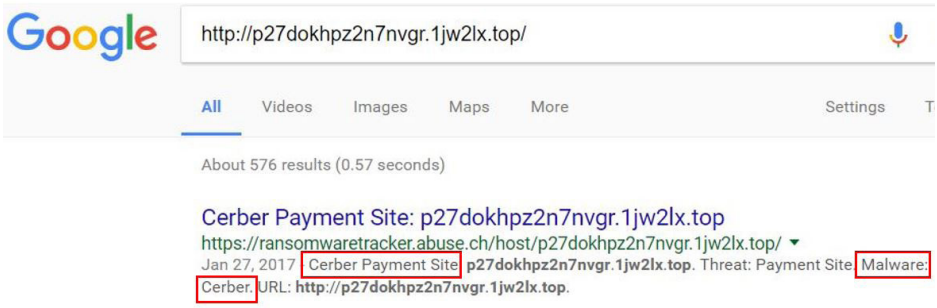


http.request

Time	Scr Port	Source	Destination	Dsc Port	Length	Host
2017-01-27 22:55:51.…	49216	172.16.4.193	194.87.234.1…	80	593	tyu.benme.com
2017-01-27 22:55:51.…	49216	172.16.4.193	194.87.234.1…	80	632	tyu.benme.com
2017-01-27 22:55:51.…	49215	172.16.4.193	194.87.234.1…	80	632	tyu.benme.com
2017-01-27 22:56:10.…	49220	172.16.4.193	198.105.121.…	80	340	p27dokhpz2n7nvgr.1jw2lx.top
2017-01-27 22:56:10.…	49220	172.16.4.193	198.105.121.…	80	350	p27dokhpz2n7nvgr.1jw2lx.top
2017-01-27 22:56:11.…	49220	172.16.4.193	198.105.121.…	80	441	p27dokhpz2n7nvgr.1jw2lx.top
2017-01-27 22:56:11.…	49220	172.16.4.193	198.105.121.…	80	414	p27dokhpz2n7nvgr.1jw2lx.top
2017-01-27 22:56:11.…	49221	172.16.4.193	198.105.121.…	80	398	p27dokhpz2n7nvgr.1jw2lx.top
2017-01-27 22:56:11.…	49222	172.16.4.193	198.105.121.…	80	441	p27dokhpz2n7nvgr.1jw2lx.top
2017-01-27 22:56:12.…	49220	172.16.4.193	198.105.121.…	80	439	p27dokhpz2n7nvgr.1jw2lx.top
2017-01-27 22:56:12.…	49221	172.16.4.193	198.105.121.…	80	435	p27dokhpz2n7nvgr.1jw2lx.top
2017-01-27 22:56:13.…	49216	172.16.4.193	194.87.234.1…	80	765	tyu.benme.com
2017-01-27 22:56:13.…	49215	172.16.4.193	194.87.234.1…	80	768	tyu.benme.com
2017-01-27 22:56:13.…	49221	172.16.4.193	198.105.121.…	80	269	p27dokhpz2n7nvgr.1jw2lx.top
2017-01-27 22:56:15.…	49223	172.16.4.193	194.87.234.1…	80	533	tyu.benme.com
2017-01-27 22:56:15.…	49221	172.16.4.193	198.105.121.…	80	445	p27dokhpz2n7nvgr.1jw2lx.top
2017-01-27 22:56:16.…	49221	172.16.4.193	198.105.121.…	80	443	p27dokhpz2n7nvgr.1jw2lx.top
2017-01-27 22:56:16.…	49221	172.16.4.193	198.105.121.…	80	433	p27dokhpz2n7nvgr.1jw2lx.top
2017-01-27 22:56:16.…	49220	172.16.4.193	198.105.121.…	80	432	p27dokhpz2n7nvgr.1jw2lx.top
2017-01-27 22:56:16.…	49222	172.16.4.193	198.105.121.…	80	527	p27dokhpz2n7nvgr.1jw2lx.top
2017-01-27 22:56:17.…	49224	172.16.4.193	198.105.121.…	80	457	p27dokhpz2n7nvgr.1jw2lx.top

Gambar 7.4 Informasi *Gathering*

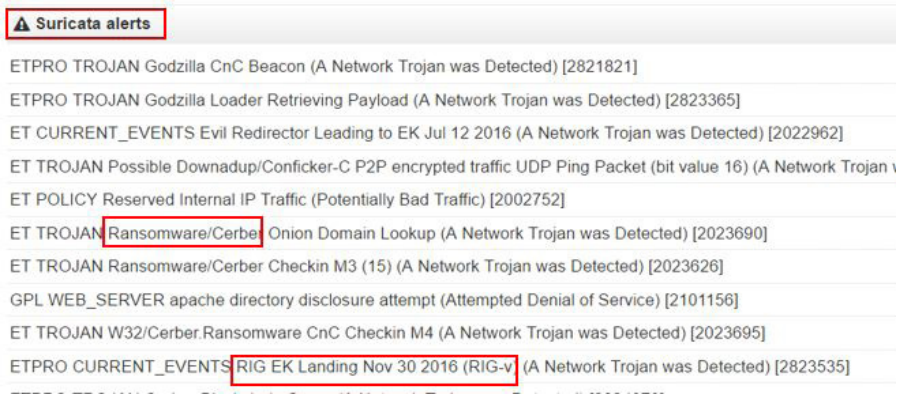
Dari hasil analisis ternyata kami menemukan bahwa domain yang digunakan oleh penjahat *cyber*, dengan bantuan mesin pencari Google dengan kata kunci p27dokhpz2n7nvgr.1jw2lx.top. Hasil pencarian Google.com menunjukkan pada Gambar 7.5, menjelaskan ditemukan *malware* yang menginfeksi PC Stewie adalah CERBER Ransomware .



Gambar 7.5 Hasil p27dokhpz2n7nvgr.1jw2lx.top

Pada Gambar 7.6, kami menampilkan hasil PCAP yang telah diunggah ke https://www.virustotal.com menunjukkan hasil Suricata

yang ditampilkan menemukan seorang aktor /cybercriminal Cerber menggunakan RIG EK (Exploit Kit).

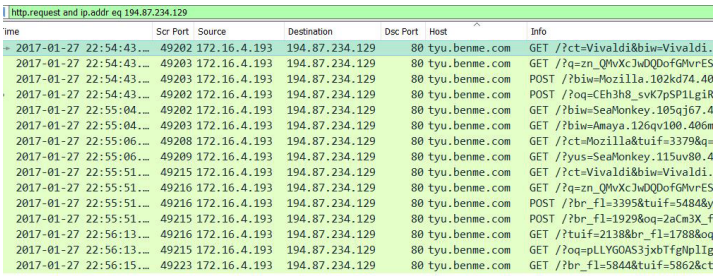


Gambar 7.6 RIG *Exploit Kit Landing*

Exploit Kit (EK) adalah kerangka kerja berbasis server, eksploitasi dengan memanfaatkan kelemahan dalam aplikasi perangkat lunak yang biasanya dikaitkan dengan *browser* web dan menginfeksi korban tanpa disadari telah terinfeksi. RIG EK adalah pengiriman dan distribusi *gateway malware* yang berfungsi mengarahkan korban untuk mengunduh sesuatu yang bermuatan *malware*.

Pada Gambar 7.7, menunjukkan hasil penyaringan http.request dan ip.addr eq 194.87.234.129 yang menunjukkan alamat IP yang terkait dengan Rig EK. Secara umum, penyebaran *ransomware* menggunakan dua metode: pertama melalui spam *mail* dan Exploit Kit. Spam *email* adalah cara penyebaran dan pendistribusian langsung ke korban *ransomware* untuk memasuki tautan yang telah terinfeksi perangkat lunak perusak dan mengambil bagian aktif pada korban untuk mengeklik tautan atau *file* lampiran yang telah disuntikkan *malware*. Metode kedua adalah menggunakan *exploits Kit*. *Exploits Kit* (EK) dirancang untuk bekerja di belakang layar, yang digunakan oleh penjahat *cyber* untuk mengotomatisasi eksploitasi lubang keamanan di mesin korban

saat melakukan penjelajahan aktif. EK tidak memerlukan tindakan aktif seperti korban klik pada link atau lampiran.



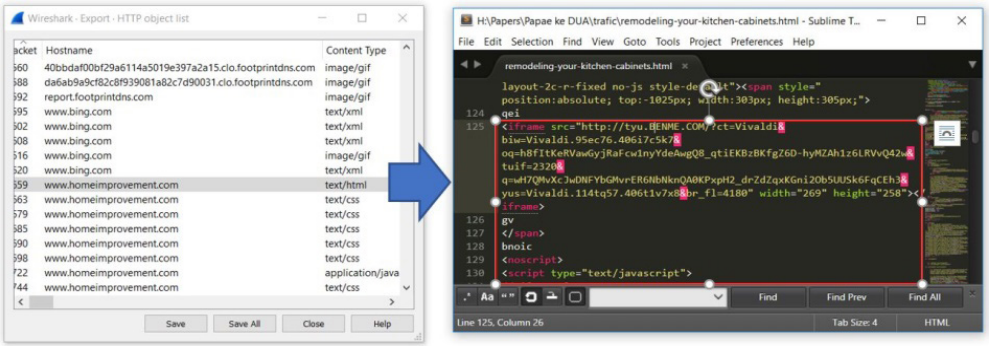
Gambar 7.7 HTTP *Requests* ke Alamat IP Rig EK

Filtering http requests pada semua alamat IP Rig EK di Wireshark, mendeteksi dan menganalisis RIG EK dan domain situs web yang menengahi penyebaran dan infeksi komputer *host* melalui TCP Stream berikut paket, seperti yang ditunjukkan pada Gambar 7.8. Setelah hasil aliran TCP menunjukkan hasil yang ditemukan komputer *host* adalah alamat www.homeimprovement.com. Dari analisis akses korban yang diketahui ke bing.com adalah melakukan pencarian dengan kata kunci *remodeling your kitchen cabinets* http://www.bing.com/search?q=home+improvement+remodeling+your+kitchen&qs=n&sp = -1 & PQ = home improvement + + + your remodeling.

Gambar 7.8 *Follow* HTTP *Stream* Filter untuk Menemukan *Referrer*

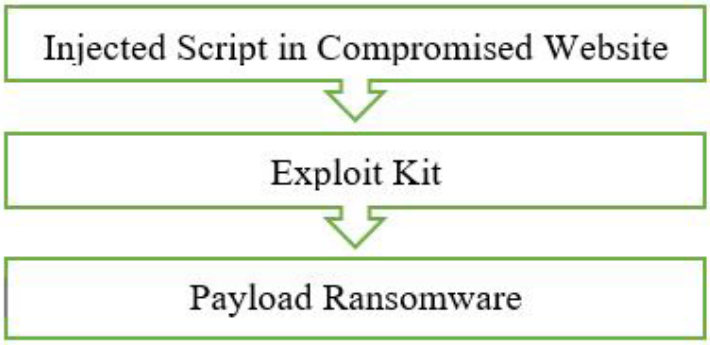
Dari hasil analisis bahwa web www.homeimprovement.com telah dikompromikan dalam penyebaran RIG EK. RIG EK adalah metode pengiriman yang canggih, sistem untuk mendistribusikan *malware* melalui EK melibatkan banyak komponen lain dalam rangkaian kejadian infeksi perangkat lunak perusak. Intinya, RIG EK dengan berbagai trik mengarahkan lalu lintas ke pengguna EK server sebelum mengirim *malware*. Pelaku memandu lalu lintas ke server korban EK.

Pelaku dan kampanye dua istilah yang berbeda, aktor dapat menggunakan satu atau beberapa kampanye untuk menyebarkan *malware*. Satu aktor mungkin telah menggunakan kampanye yang sama untuk mendistribusikan berbagai jenis perangkat lunak perusak. Tahap selanjutnya adalah menentukan *script* kampanye yang digunakan untuk mengantarkan *cerber* adalah cara untuk mengekspor objek dalam *capture* paket, seperti yang ditunjukkan pada Gambar 7.9.



Gambar 7.9 Ekspor Daftar Objek dan Skrip PseudoDarkleech

PseudoDarkleech adalah sebuah kode/*script* yang dipakai oleh pembuat *cerber ransomware*, berfungsi untuk mengarahkan lalu lintas dari korban ke server Exploit Kit dengan mode diam-diam. *Script* pseudodarkleech memiliki tugas untuk menyuntikkan ke halaman web dan server web sampai ke tingkat root/.



Gambar 7.10 Chain Event Pseudo-Darkleech

Penjelasan rangkaian kejadiannya yang ditunjukkan pada Gambar 7.10 adalah sebagai berikut:

1. Korban pertama mengunjungi situs web (situs web yang disusupi) yang telah disusupi atau *skrip* berbahaya, dan *skrip* disuntikkan ke situs web yang dikompromikan untuk mengajukan permintaan HTTP pada Exploit Kit *landing page*.
2. *Landing page* EK menemukan dan menentukan apakah komputer memiliki kerentanan biasanya aplikasi berbasis *browser* dan Adobe flash player dan selanjutnya mengirim EK Exploit untuk memanfaatkan aplikasi yang rentan.
3. Jika eksploitasi berhasil, EK mengirimkan *payload ransomware* dan melakukan aktivitas untuk mengakses dan mengenkripsi *file* dan folder tanpa disadari, korban benar-benar telah terinfeksi oleh *payload ransomware*.

7.2 Soal *Traffic Analysis Exercise*

Ilustrasinya, Anda bekerja sebagai analis keamanan untuk sebuah perusahaan dengan lokasi di seluruh dunia. Baru-baru ini di tempat Anda berkerja telah membuka kantor cabang di Jepang. Suatu hari di

hari Selasa, 2017-06-27, Anda melihat beberapa peringatan prioritas tinggi dari dua *intrusion detection systems* (IDS) yang berbeda. Satu IDS menjalankan Snort menggunakan aturan Snort Ruleset, dan yang lainnya menjalankan Suricata menggunakan aturan EmergingThreats Pro. (*File* pcap silakan Anda *download* di https://adekurniawanrusdy.wordpress.com/)

Hasilnya menunjukkan bahwa komputer berbasis Windows di kantor cabang Jepang telah terinfeksi. Oleh atasan Anda ditugaskan untuk menyelidikinya! Anda memiliki pcap, *file* yang berisi Snort *alert*, dan *file* yang berisi Suricata *alert*.

Untuk *case study* analisis *network forensics*-nya, di bawah ini merupakan tugas yang Anda harus lakukan:

1. Berapakah MAC *address*, IP *address*, dan *host name* dari komputer Windows terinfeksi?
2. Berapakah tanggal dan waktu (dalam UTC) komputer terinfeksi?
3. Berdasarkan Snort dan Suricata *alerts*, komputer apa yang terinfeksi?
4. Berdasarkan indikator dari permintaan HTTP GET pertama, tentukan bagaimana komputer terinfeksi?
5. Berdasarkan jawaban sebelumnya, berapakah hash SHA256 untuk *file* yang mungkin menginfeksi komputer?
6. File pcap berisi tiga *file executable* Windows yang dikirim melalui HTTP. Ekspor mereka dari pcap. Temukan *file hash* SHA256 dari ketiga *file* tersebut?

Bab 8
Forensika Awan/ *Cloud Forensics*

Tujuan Instruksional Umum:

1. Mahasiswa/wi mampu memahami *cloud forensics*.

Tujuan Instruksional Khusus:

1. Mahasiswa/wi mampu memahami hal dasar tentang *cloud computing* dan *cloud forensics*.
2. Mahasiswa/wi mampu mengenal dan memahami tantangan dalam *cloud forensics*.
3. Mahasiswa/wi mampu mahasiswa/wi mampu mengenal dan memahami peluang dalam *cloud forensics*.

8.1 *Cloud Computing* dan *Cloud Forensics*

Lingkungan komputer modern telah bergerak melewati *data center* lokal dengan satu titik masuk dan keluar ke jaringan global yang terdiri dari banyak *data center*. Migrasi bisnis bergerak cepat, di mana layanan *data center* jarak jauh, komputasi dan penyimpanan disewa dari perusahaan yang lebih besar, disebut sebagai komputasi awan (*cloud*

computing). Perusahaan dan orang-orang telah menyadari manfaat besar yang dihasilkan dari penggunaan sistem komputasi awan, tidak hanya dalam hal produktivitas, tetapi juga dalam akses ke sistem berkecepatan tinggi untuk mengelola kumpulan data yang sangat besar yang secara finansial mustahil bagi beberapa perusahaan kecil dan menengah. Perusahaan besar juga telah menyadari manfaat murah dari *cloud computing*, migrasi dari sistem pengolahan tradisional, dan paket penyewaan perangkat lunak di *data center* yang dapat diakses di mana saja di dunia. Migrasi ini juga memiliki komplikasi pada keamanan informasi karena harus memahami proses keamanan informasi, baik secara prosedural dan legal.

Sistem komputer dan forensik jaringan dipengaruhi oleh perubahan dari *data center* lokal ke *data center* jarak jauh, di mana akses tidak memungkinkan secara fisik. Virtualisasi juga telah mengubah sifat komputer keamanan dan komputer forensik dalam hubungan dengan bagaimana komputer dilihat, ketika berhadapan dengan insiden keamanan yang sebenarnya. Ini berarti bahwa akan terus ada perubahan dalam bagaimana keamanan komputer dan penyelidikan forensik selesai, ketika beberapa atau semua sistem tidak dapat diakses secara fisik.

Mungkin kita berpikir sekarang bahwa satu perangkat fisik hanya akan memiliki satu sistem operasi yang akan diinvestigasi. Di *cloud computing*, server fisik dapat memiliki banyak server virtual yang berjalan pada perangkat keras fisik dan server virtual tersebut mungkin bahkan bukan milik perusahaan atau layanan yang sama.

Cloud computing akan memerlukan perubahan pada kebijakan perusahaan dan keamanan terkait akses jarak jauh dan penggunaan data melalui *browser*, privasi, mekanisme audit, sistem pelaporan, dan sistem manajemen yang menggabungkan bagaimana data diamankan pada sistem komputer sewaan yang dapat berada di mana saja. Ini adalah konteks penuh dari sistem komputasi awan yang digunakan perusahaan

yang membuat lingkungan keamanan yang kompleks dan menantang dan yang menentukan perimeter keamanan modern. Lingkar keamanan sekarang harus dilihat sebagai serangkaian sistem (paket perangkat keras dan sistem operasi dalam lingkungan virtual), data, aturan akses dan kebijakan yang mengatur data dan akses, serta respons insiden yang hanya cenderung menyulitkan arsitektur dan proses pendukung. Ini memerlukan pendekatan yang sama sekali baru untuk tidak hanya bagaimana sistem diprogram, tetapi juga bagaimana keamanan informasi dilakukan. Perubahan ini belum ditangani secara baik, meskipun penyedia layanan *cloud* yang lebih besar mulai memenuhi kebutuhan industri. Seiring waktu, ini akan mencakup bagaimana perusahaan benar-benar dapat menangani forensik jaringan dan komputer dalam lingkungan komputasi awan.

8.2 *Cloud Forensics*

Forensik jaringan dalam lingkungan komputasi awan dapat difokuskan hanya pada data yang masuk dari dan ke sistem yang dapat diakses oleh perusahaan. Forensik jaringan perlu menjadi bagian bekerja dari semua komponen lain yang membentuk seluruh sistem dalam lingkungan *cloud*. Tanpa penyelidik forensik jaringan, memahami arsitektur sistem lingkungan *cloud* dan kemungkinan kompromi akan diabaikan atau dilewatkan. Penyelidik forensik jaringan juga perlu memahami bahwa lingkungan *cloud* adalah ruang yang disewakan perusahaan pada sistem komputer perusahaan lain untuk melakukan pekerjaan. Ruang yang disewa di awan dapat berada di pusat data yang terhubung secara global dengan banyak perusahaan lain di mana titik masuk jaringan pengguna dapat berada di titik mana pun di internet. Data di lingkungan *cloud* dapat direplikasi ke pusat data apa pun di dunia yang dimiliki dan dioperasikan oleh penyedia *cloud*. Penyedia *cloud* memiliki serangkaian kebijakan, sistem keamanan, perangkat keras,

dan paket perangkat lunak yang independen dari apa yang dilakukan perusahaan di ruang *cloud*. Pelanggan *cloud computing* mungkin atau tidak memiliki akses ke data yang berhubungan dengan mereka secara khusus jika komputer diduga telah dikompromikan oleh peretas atau jika data dicuri oleh orang dalam atau orang luar.

Antara penyedia *cloud* dan konsumen *cloud* ini memberikan lahan subur bagi para peretas dan kriminal yang ingin meretas sistem untuk tujuan mereka sendiri. Ini juga menyediakan lahan subur bagi orang dalam juga karena biaya untuk menyiapkan komputer *cloud* sangat murah. Dengan sekitar Rp560.000 per bulan, server *cloud* lengkap dapat diatur untuk digunakan untuk tujuan apa pun oleh siapa pun dengan kartu kredit. Program sederhana seperti WinSCP dapat digunakan untuk mengakses komputer awan itu, atau jika dikonfigurasi, itu bisa saja seperti server *file transfer protocol* (FTP) lainnya di internet yang berarti bahwa setiap klien FTP termasuk proses pemasangan Windows dapat digunakan untuk menjatuhkan data di server *cloud*. Beberapa perusahaan seperti Dropbox dan Mozy menawarkan layanan ini secara gratis hingga 2 GB informasi per alamat *e-mail* pengguna.

Biaya untuk melakukan forensik jaringan dalam lingkungan komputasi awan dapat menghancurkan sebuah perusahaan jika data mereka hilang atau dicuri oleh seorang karyawan. Komputasi awan, dengan aset dan keterbatasannya, juga bisa menjadi lingkungan yang sulit bagi para profesional keamanan informasi yang terlatih untuk memahami betapa forensik jaringan itu dan bagaimana digital forensik tradisional tidak sepenuhnya masuk ke dalam lingkungan komputasi awan.

Cloud computing dapat dianggap sebagai penyewaan ruang komputer di pusat data perusahaan lain. Ini menyiratkan bahwa perusahaan memiliki kontrol atas beberapa aspek sistemnya tergantung pada layanan *cloud* mana yang dibeli/sewa oleh perusahaan. Namun, ada ditemukan

kurangnya kontrol terhadap sistem komputasi perusahaan di data *center* tradisional atau lingkungan komputasi. Perubahan diperlukan dalam cara perusahaan menangani keamanan informasi melalui kontrol, kebijakan, dan solusi teknis karena kontrol total aset komputasi dan jaringan tidak mungkin dalam lingkungan komputasi awan. Secara pragmatis, dalam komputasi awan, perusahaan hanya membeli mesin virtual di pusat data orang lain.

Penyedia layanan *cloud* juga memiliki serangkaian kekuatan dan kelemahan yang melekat yang muncul dengan filosofi desain yang dimiliki penyedia layanan *cloud* ketika merancang sistemnya. Keputusan desain dan arsitektur pada bagian dari penyedia layanan *cloud* memberikan batasan pada apa yang dapat dan tidak dapat dilakukan dalam analisis forensik. Penting bahwa penyelidik forensik memahami pertimbangan desain ini yang masuk ke dalam arsitektur penyedia layanan *cloud*. Amazon, Rackspace, dan Microsoft Azure semuanya memiliki filosofi desain yang sangat berbeda, mereka menyediakan layanan komputasi awan yang akan mempersulit proses forensik jaringan apa pun.

Ambil suatu contoh *Amazon Web Services* (AWS), dengan *Amazon Machine Image* (AMI) dengan sistem operasi Linux atau Windows. Anda dapat menjalankan mesin virtual itu dan melakukan apa pun yang ingin kita lakukan dengannya. Anda tidak memiliki infrastruktur jaringan, dan Anda tidak memiliki *firewall* di pusat data, Anda juga tidak memiliki perangkat keras pendukung di bawah sistem operasi. Namun, Anda memiliki seluruh mesin virtual, Linux atau Windows, dan dapat melakukan apa pun yang Anda inginkan dalam batasan sistem virtual itu. Ini adalah pengaturan yang sama dengan yang dimiliki perusahaan dalam sistem virtual mereka sendiri di pusat data yang dikendalikan perusahaan mereka sendiri. Ini juga membuat migrasi alat dan aplikasi lebih mudah untuk alat keamanan tradisional yang perlu membuat perubahan pada *registri* sistem komputer untuk beroperasi.

Layanan platform dan *hosting* yang dibeli oleh perusahaan untuk *cloud computing* merupakan titik keputusan penting bagi forensik jaringan. Ketika membuat keputusan tentang penyedia layanan apa yang digunakan, penting juga untuk memahami cara kerja komputasi awan, apa yang dapat dilakukan dengan itu, dan apa yang tidak dapat dilakukan dengan komputasi awan. Beberapa proses akan menjadi sangat baik dalam lingkungan komputasi awan, seperti pemrosesan layanan transaksi.

Keterbatasan yang melekat pada komputasi awan juga perlu dipahami, jika ingin proses forensik jaringan dan komputer berhasil dalam lingkungan ini. Keputusan untuk menggunakan penyedia layanan *cloud* harus ditinjau tidak hanya dalam hal layanan apa yang ditawarkan layanan *cloud*, tetapi juga dalam hal bagaimana perusahaan membeli layanan komputasi awan memutuskan untuk menggunakannya. Keputusan-keputusan ini memiliki implikasi langsung pada bagaimana jaringan dan sistem forensik akan dilakukan. Penting bahwa setiap perusahan mempunyai departemen yang memiliki layanan dan mampu membangun layanan keamanan serta juga layanan pemantauan berdasarkan penyedia layanan *cloud*.

Namun, terlepas dari penyedia, lingkungan virtual akan menyulitkan, dan dalam beberapa kasus, itu akan mengurangi efektivitas forensik berbasis jaringan. Persamaan penyedia layanan *cloud* adalah sebagai berikut:

- Tidak ada akses ke *router* jaringan, *load balancing*, atau komponen berbasis jaringan lainnya.
- Tidak ada akses ke instalasi *firewall*.
- Tidak ada kemampuan yang sebenarnya untuk merancang peta jaringan *hops* yang diketahui dari satu instansi ke lainnya yang akan tetap statis atau konsisten di seluruh skema perutean.

- Sistem menjadi sistem komoditas karena dirancang untuk dibangun dan diruntuhkan sesuka hati. Ketika mesin virtual (VM) diruntuhkan, tidak ada data fisik dan itu hilang begitu saja. Jika VM pernah *shutdown* maka seluruh sistem termasuk log juga dapat dihancurkan dan tidak pernah pulih.
- VM akan dibangun dan diruntuhkan oleh sejumlah administrator sistem administrator di sebuah perusahaan sebagai layanan sesuai permintaan, perusahaan harus membuat seluruh rangkaian kebijakan keamanan baru dan berencana untuk bekerja dengan server *cloud* yang dikompromikan.

Konsep forensik jaringan dalam komputasi awan membutuhkan pola pikir baru di mana beberapa data tidak akan tersedia, beberapa data akan dicurigai, dan beberapa data akan siap untuk pengadilan dan dapat sesuai dengan jaringan tradisional. Tantangan bagi penyelidik forensik adalah untuk memahami kumpulan data apa yang terkumpul yang jatuh ke dalam masing-masing kategori yang tidak tersedia. Bekerja dengan penasihat hukum perusahaan dan ahli *cloud computing* akan menjadi kebutuhan.

Model komputasi awan juga dapat sangat berguna untuk forensik dengan memungkinkan penyimpanan *file log* yang sangat besar pada penyimpanan basis data yang sangat besar untuk penemuan barang bukti akan berlangsung sangat mudah. Selain itu, beberapa alat visualisasi data *cloud* yang terbaru terdapat fitur forensik dan peringatan dini yang sangat baik bagi para administrator keamanan dan penyelidik keamanan. Beberapa alat visualisasi data *cloud* seperti NetFlow. NetFlow tidak pernah dimaksudkan untuk menjadi alat forensik jaringan. Insinyur keamanan dan pekerja IT diharuskan untuk berkreasi dengan set alat mereka saat ini dan membuatnya berfungsi di lingkungan *cloud*. Masalah tambahan tentang bagaimana penyelidikan forensik jaringan dapat berhasil dengan *cloud* adalah bahwa komputasi awan adalah lingkungan

yang tidak dikenal bagi para insinyur keamanan. Departemen keamanan harus menjadi bagian dari seluruh proses pengambilan *cloud* dari arsitektur ke layanan dan sistem yang akan dimasukkan ke dalam pusat data penyedia layanan *cloud*.

Departemen keamanan harus tahu bagaimana *cloud* dan layanan internal saling terkait satu sama lain, serta bagaimana arsitektur dirancang untuk mengakomodasi pembagian data di berbagai batasan dan lapisan komputasi. Komputasi awan dilengkapi dengan seperangkat standar, terminologi, dan praktik terbaiknya sendiri yang sulit untuk dikelola dalam konteks keamanan informasi tradisional. Hal ini juga sangat sulit, saat ini, untuk mengetahui apa yang terjadi pada sistem di lingkungan komputasi awan karena kurangnya alat yang dikembangkan dengan baik atau standar dan praktik keamanan informasi.

Glosarium

Advanced Research Projects Agency (DARPA):

Agen dari Departemen Pertahanan A.S. yang bertanggung jawab atas pengembangan teknologi baru untuk digunakan oleh militer.

Botnet:

Sekelompok komputer yang terhubung secara terkoordinasi untuk tujuan jahat.

***Buffer overflows*:**

Terjadi ketika lebih banyak data ditulis ke *buffer* daripada yang dapat ditahan.

***Circuit switching*:**

Metode untuk mengimplementasikan jaringan telekomunikasi di mana dua node jaringan membentuk saluran komunikasi khusus (*circuit*) yang menjamin saluran *bandwidth* tetap penuh dan terhubung selama sesi komunikasi.

***Cloud computing*:**

Istilah umum untuk layanan *host* melalui internet.

***Command control* (C2):**

Komputer terpusat yang mengeluarkan perintah ke botnet.

Cross Site Scripting (XSS):

Mengacu pada serangan injeksi kode di sisi klien d imana penyerang dapat mengeksekusi *skrip* berbahaya (juga biasa disebut muatan berbahaya) ke situs web atau aplikasi web yang sah.

Denial-of-service (DoS):

Membanjiri server, sistem atau jaringan lalu lintas korban, sehingga membuat sulit atau tidak mungkin bagi pengguna yang sah untuk menggunakannya**.**

***Digital subscriber line* (DSL):**

Teknologi yang digunakan untuk mentransmisikan data digital melalui saluran telepon.

***Domain name system* (DNS):**

Sistem internet untuk mengubah nama alfabet menjadi alamat IP numerik.

***Graphical user interface*:**

Antarmuka pengguna yang memungkinkan pengguna berinteraksi dengan perangkat elektronik melalui ikon grafis dan indikator visual.

***Grid computing*:**

Arsitektur prosesor yang menggabungkan sumber daya komputer dari berbagai domain.

***Honeypot*:**

Mekanisme keamanan komputer yang diatur untuk mendeteksi, membelokkan, atau, dengan cara tertentu, menangkal upaya penggunaan sistem informasi yang tidak sah.

***Internet control message protocol* (ICMP)**

Protokol pelaporan kesalahan, digunakan untuk menghasilkan pesan kesalahan ke alamat IP sumber saat masalah jaringan terjadi.

Injeksi SQL:

Teknik injeksi kode, yang digunakan untuk menyerang aplikasi berbasis data, di mana pernyataan SQL jahat dimasukkan ke dalam *field entri* untuk dieksekusi.

***Network control protocol* (NCP)**

Protokol awal yang diterapkan oleh ARPANET, jaringan *packet switching operasional* pertama di dunia yang kemudian berkembang menjadi apa yang kenal sekarang bernama internet.

***Packet capture*:**

Jaringan komputer untuk mencegat paket data yang melintasi atau berpindah melalui jaringan komputer tertentu.

Pastebin:

Situs web tempat Anda dapat menyimpan teks secara *online* agar mudah dibagikan. Situs ini terutama digunakan oleh pemrogram untuk menyimpan beberapa kode sumber atau informasi konfigurasi.

***Peer-to-Peer*:**

Model jaringan di mana dua atau lebih komputer membentuk hubungan jaringan tidak terstruktur dan tidak resmi.

***Port*:**

Mekanisme yang mengizinkan sebuah komputer untuk mendukung beberapa sesi koneksi dengan komputer lainnya dan program di dalam jaringan.

Prinsip *end-to-end*:

Kerangka desain dalam jaringan komputer, yang memiliki aplikasi fitur khusus berada di node jaringan yang berkomunikasi, bukan di node perantara, seperti *gateway* dan *router*.

***Pretty good privacy* (PGP)**

Program yang digunakan untuk mengenkripsi dan mendekripsi *e-mail* melalui internet.

***Programmable logic controllers* (PLC):**

Sistem kontrol komputer industri yang terus memantau keadaan perangkat *input* dan membuat keputusan berdasarkan program kustom untuk mengendalikan keadaan perangkat *output*.

***Rootkit*:**

Kumpulan alat (program) yang memungkinkan akses tingkat administrator ke komputer atau jaringan komputer.

***Secure sockets layer* (SSL**)

Protokol aplikasi jaringan komputer untuk mengamankan koneksi antara klien dan server melalui jaringan yang tidak aman, seperti internet.

***Server message block* (SMB):**

Protokol *client*/server yang ditujukan sebagai layanan untuk berbagi berkas (*file sharing*) di dalam sebuah jaringan.

***Simple mail transfer protocol* (SMTP)**

Protokol TCP/IP yang digunakan dalam mengirim dan menerima *e-mail*.

(Lillard, Garrison, Schiller, & Steele, 2010)

***Supervisory control and data acquisition systems* (SCADA):**

Sistem industri yang dikendalikan komputer yang penting yang terus memantau dan mengendalikan berbagai bagian infrastruktur industri.

***Transport layer security* (TLS)**

Protokol aplikasi yang menyediakan integritas privasi dan data yang sedang berkomunikasi.

VoIP *over wireless* (VoIPoW):

Pengangkutan suara melalui jaringan berbasis IP.

Wi-Fi:

Versi nirkabel dari jaringan Ethernet kabel.

World Wide Web:

Sistem informasi di internet yang memungkinkan dokumen dihubungkan ke dokumen lain dengan tautan *hypertext*, memungkinkan pengguna mencari informasi dengan berpindah dari satu dokumen ke dokumen lainnya.

***Zero-day vulnerability*:**

Cacat pada perangkat lunak, perangkat keras atau *firmware* yang tidak diketahui oleh pihak atau pihak yang bertanggung jawab untuk menambal atau memperbaiki kekurangannya.

(Adamov & Carlsson, 2017; Casey, 2011; Kurniawan, Riadi, & Luthfi, 2017; Provataki & Katos, 2013; Riadi, Eko, Ashari, & Subanar, 2013)(Davidoff & Ham, 2012)(Kurose James F.; Ross, 2013)(J. M. Kizza, 2013; Lehtinen & Sr, n.d.; Vacca, 2009)(Alpaydin, 2013; Budiharto, 2016; Norvig & Russell, 2010; Sebe, Cohen, Garg, & Huang, 2005) (Casey, 2011; Eoghan casey, 2010; Eoghan Casey, n.d.)(Kurniawan & Riadi, 2018)(Jacobson, 2008)(M. Kizza, David, & Sc, 2009; Pawar & Anuradha, 2015)(Response, n.d.)(Messier, 2017)(Chuvakin, 2003; Daimi, 2017; Ii & Heiser, 2001; Stallings, 2011)(Lillard et al., 2010) (Pilli, Joshi, & Niyogi, 2011)(Stallings, 2005, 2011)(Daimi, 2017) (Carvey, n.d.; Sammons, 2012)(Daniel & Daniel, 2012)(Bosworth, Kabay, & Whyne, 2014)(Daimi, 2017)(Konheim, 2006).

Daftar Pustaka

Adamov, A., & Carlsson, A. (2017). The State of Ransomware. Trends and Mitigation Techniques. In *2017 IEEE East-West Design & Test Symposium (EWDTS)* (pp. 1–8). IEEE. https://doi.org/10.1109/EWDTS.2017.8110056

Alpaydin, E. (2013). *Introduction to Machine Learning* (Vol. 53). MIT Press.

Bosworth, S., Kabay, M. E., & Whyne, W. (2014). *Computer Security Handbook*. *Computer Law & Security Review* (Vol. 22). https://doi.org/10.1016/j.clsr.2005.12.007

Budiharto, W. (2016). *Machine Learning & Computational Intelligence*. Yogyakarta: Andi Offset.

Carvey, C. A. H. (n.d.). *Digital Forensics with Open Source Tools*.

Casey, E. (2011). *Digital Evidence and Computer Crime*. (E. Casey, Ed.) (Third Edit). Maryland: Elsevier Academic Press. https://doi.org/10.1017/CBO9781107415324.004

Chuvakin, A. (2003). Honeypot Essentials. *Information Systems Security*, *11*(6), 15–20. https://doi.org/10.1201/1086/43324.11.6.20030101/40427.4

Daimi, K. (2017). *Computer and Network Security Essentials*. *Computer and Network Security Essentials*. https://doi.org/10.1007/978-3-319-58424-9

Daniel, L., & Daniel, L. (2012). *Digital Forensics for Legal Professionals*. *Digital Forensics for Legal Professionals*. https://doi.org/10.1016/C2010-0-67122-7

Davidoff, S., & Ham, J. (2012). *Network Forensics: Tracking Hackers Through Cyberspace*. https://doi.org/10.1017/CBO9781107415324.004

Eoghan casey. (2010). *Handbook of Digital Forensics and Investigation*. Elsevier Academic Press.

Eoghan Casey. (n.d.). *HANDBOOK OF COMPUTER CRIME INVESTIGATION*.

Ii, W. G. K., & Heiser, J. G. (2001). Computer Forensics: Incident Response Essentials (Google eBook). In *Computer Forensics: Incident Response Essentials* (Vol. 7, p. 416). https://doi.org/10.1109/MSP.2005.95

Jacobson, D. (2008). Introduction to Network Security, *18*, 504. Retrieved from https://books.google.com/books?id=50DMBQAAQBAJ&pgis=1

Kizza, J. M. (2013). *Computer Communication and Network*. *Computer Network Security*. https://doi.org/10.1007/978-1-4471-4543-1

Kizza, M., David, K., & Sc, M. (2009). "Guide to Computer Network Security" What the book is about, (Iso 17799), 1–5.

Konheim, A. G. (2006). *Computer Security and Cryptography*. *Computer Security and Cryptography*. https://doi.org/10.1002/0470083980

Kurniawan, A., & Riadi, I. (2018). Detection and Analysis Cerber Ransomware Based on Network Forensics Behavior. *International Journal of Network Security*, *20*(3), 1–7.

Kurniawan, A., Riadi, I., & Luthfi, A. (2017). Forensic Analysis and Prevent of Cross Site Scripting in Single Victim Attack Using Open Web Application Security Project (OWASP) Framework. *Journal of Theoretical and Applied Information Technology*, *95*(6), 1363–1371.

Kurose James F.; Ross, K. W. (2013). *Computer Networking : a Top-Down Approach*.

Lehtinen, R., & Sr, G. T. G. (n.d.). Computer Security Basics, 2nd Edition (2006).

Li, C. (n.d.). *Handbook of Research on Computational Forensics, Digital Crime, and Investigation: Methods and Solutions*.

Lillard, T. T., Garrison, C. C., Schiller, C. C., & Steele, J. (2010). *Digital Forensics for Network, Internet, and Cloud Computing*. *Digital Forensics for Network, Internet, and Cloud Computing*. https://doi.org/10.1016/C2009-0-62467-3

Messier, R. (2017). *Network Forensics*. Indianapolis, Indiana: John Wiley & Sons, Inc. https://doi.org/10.1002/9781119329190

Norvig, P., & Russell, S. J. (2010). *Artificial Intelligence: A Modern Approach*. (P. Norvig & S. J. Russell, Eds.). Pearson Prentice Hall.

Pawar, M. V., & Anuradha, J. (2015). Network Security and Types of Attacks in Network. *Procedia Computer Science*, *48*(C), 503–506. https://doi.org/10.1016/j.procs.2015.04.126

Pilli, E., Joshi, R., & Niyogi, R. (2011). *Router and Interface Marking for Network Forensics*. *Advances in Digital Forensics VII*.

Provataki, A., & Katos, V. (2013). Differential Malware Forensics. *Digital Investigation*, *10*(4), 311–322. https://doi.org/10.1016/j.diin.2013.08.006

Response, C. I. (n.d.). Network Forensics and Analysis Poster SOF-ELK Network-Based Distilling Full-Packet.

Riadi, I., Eko, J., Ashari, A., & Subanar. (2013). Log Analysis Techniques Using Clustering in Network Forensics. *International Journal of Computer Science and Information Security*, *Vol. 10*(August 2016).

Sammons, J. (2012). *The Basics of Digital Forensics*. *The Basics of Digital Forensics*. https://doi.org/10.1016/C2010-0-68337-4

Sebe, N., Cohen, I. R. A., Garg, A., & Huang, T. S. (2005). *Machine Learning in Computer Vision*. *New York*.

Stallings, W. (2005). *Cryptography and Network Security: Principles and Practices*. *Cryptography and Network Security*. https://doi.org/10.1007/11935070

Stallings, W. (2011). *Network Security Essentials: Applications and Standards*. *William Stallings Books on Computer and Data Communications Technology*. https://doi.org/William Stallings Books on Computer and Data Communications Technology

Symantec. (2017). Internet Security Threat Report 2017, (April), 1–77. Retrieved from https://www.symantec.com/content/dam/symantec/docs/reports/istr-22-2017-en.pdf

Tanenbaum, A. S. (1996). *Computer Networks*. *World Wide Web Internet And Web Information Systems* (Vol. 52). https://doi.org/10.1016/j.comnet.2008.04.002

Vacca, J. R. (2009). *Computer and Information Security Handbook*. Elsevier Inc.