KEAMANAN DARI DEVIL PROGRAM

Taksonomi ancaman perangkat lunak / klasifikasi program jahat (malicious program):

1. Program-program yang memerlukan program inang (host program). Fragmen program tidak dapat mandiri secara independen dari suatu program aplikasi, program utilitas atau program sistem.
2. Program-program yang tidak memerlukan program inang. Program sendiri yang dapat dijadwalkan dan dijalankan oleh sistem operasi.

Tipe-tipe program jahat :

1. **Bacteria** : program yang mengkonsumsi sumber daya sistem dengan mereplikasi dirinya sendiri. Bacteria tidak secara eksplisit merusak file. Tujuan program ini hanya satu yaitu mereplikasi dirinya. Program bacteria yang sederhana bisa hanya mengeksekusi dua kopian dirinya secara simultan pada sistem multiprogramming atau menciptakan dua file baru, masing-masing adalah kopian file program bacteria. Kedua kopian in kemudian mengkopi dua kali, dan seterusnya.

2. **Logic bomb** : logik yang ditempelkan pada program komputer agar memeriksa suatu kumpulan kondisi di sistem. Ketika kondisi-kondisi yang dimaksud ditemui, logik mengeksekusi suatu fungsi yang menghasilkan aksi-aksi tak diotorisasi.
   - Logic bomb menempel pada suatu program resmi yang diset meledak ketika kondisi-kondisi tertentu dipenuhi.
   - Contoh kondisi-kondisi untuk memicu logic bomb adalah ada atau tudak adanya file-file tertentu, hari tertentu daru minggu atau tanggal, atau pemakai menjalankan aplikasi tertentu. Begitu terpicu, bomb mengubah atau menghapus data atau seluruh file, menyebabkan mesin terhenti, atau mengerjakan perusakan lain.

3. **Trapdoor :** Titik masuk tak terdokumentasi rahasia di satu program untuk memberikan akses tanpa metode-metode otentifikasi normal.
   - Trapdoor telah dipakai secara benar selama bertahun-tahun oleh pemogram untuk mencari kesalahan program. Debugging dan testing biasanya dilakukan pemogram saat mengembangkan aplikasi. Untuk program yang mempunyai prosedur otentifikasi atau setup lama atau memerlukan pemakai memasukkan nilai-nilai berbeda untuk menjalankan aplikasi maka debugging akan lama bila harus melewati prosedur-prosedur tersebut. Untuk debug program jenis ini, pengembang membuat kewenangan khusus atau menghilangkan keperluan setup dan otentifikasi.
   - Trapdoor adalah kode yang menerima suatu barisan masukan khusus atau dipicu dengan menjalankan ID pemakai tertentu atau barisan kejahatan tertentu. Trapdoor menjadi ancaman ketika digunakan pemrogram jahat untuk memperoleh pengkasesan tak diotorisasi.
   - Pada kasus nyata, auditor (pemeriks) perangkat lunak dapat menemukan trapdoor pada produk perangkat lunak dimana nama pencipta perangkat lunak berlakuk sebagai password yang memintas proteksi perangkat lunak yang dibuatnya. Adalah sulit mengimplementasikan kendali-kendali perangkat lunak untuk trapdoor.

4. **Trojan horse** : Rutin tak terdokumentasi rahasia ditempelkan dalam satu program berguna. Program yang berguna mengandung kode tersembunyi yang ketika dijalankan melakukan suatu fungsi yang tak diinginkan. Eksekusi program menyebabkan eksekusi rutin rahasia ini.
   - Program-program trojan horse digunakan untuk melakukan fungsi-fungsi secara tidak langsung dimana pemakai tak diotorisasi tidak dapat melakukannya secara langsung. Contoh, untuk dapat mengakses file-file pemakai lain pada sistem dipakai bersama, pemakai dapat menciptakan program trojan horse.

- Trojan horse ini ketika program dieksekusi akan mengubah ijin-ijin file sehinga file-file dapat dibaca oleh sembarang pemakai. Pencipta program dapat menyebarkan ke pemakai-pemakai dengan menempatkan program di direktori bersama dan menamai programnya sedemikian rupa sehingga disangka sebagai program utilitas yang berguna.
- Program trojan horse yang sulit dideteksi adalah kompilator yang dimodifikasi sehingga menyisipkan kode tambahan ke program-program tertentu begitu dikompilasi, seperti program login. Kode menciptakan trapdoor pada program login yang mengijinkan pencipta log ke sistem menggunakan password khusus. Trojan horse jenis ini tak pernah dapat ditemukan jika hanya membaca program sumber. Motivasi lain dari trojan horse adalah penghancuran data. Program muncul sebagai melakukan fungsi-fungsi berguna (seperti kalkulator), tapi juga secara diam-diam menghapus file-file pemakai.
- Trojan horse biasa ditempelkan pada program-program atau rutin-rutin yang diambil dari BBS, internet, dan sebagainya.

5. **Virus** : Kode yang ditempelkan dalam satu program yang menyebabkan pengkopian dirinya disisipkan ke satu program lain atau lebih, dengan cara memodifikasi program-program itu.
   - Modifikasi dilakukan dengan memasukkan kopian program virus yang dapat menginfeksi program-program lain. Selain hanya progasi, virus biasanya melakukan fungsi yang tak diinginkan.
   - Di dalam virus komputer, terdapat kode intruksi yang dapat membuat kopian sempurna dirinya. Ketika komputer yang terinfeksi berhubungan (kontak) dengan perangkat lunak yang belum terinfeksi, kopian virus memasuki program baru. Infeksi dapat menyebar dari komputer ke komputer melalui pemakai-pemakai yang menukarkan disk atau mengirim program melalui jaringan. Pada lingkungan jaringan, kemampuan mengakses aplikasi dan layanan-layanan komputer lain merupakan fasilitas sempurna penyebaran virus.
   - Masalah yang ditimbulkan virus adalah virus sering merusak sistem komputer seperti menghapus file, partisi disk, atau mengacaukan program.
   - **Siklus hidup Virus** melalui empat fase (tahap), yaitu :
     - ⇒ Fase tidur (dormant phase). Virus dalam keadaan menganggur. Virus akan tiba-tiba aktif oleh suatu kejadian seperti tibanya tanggal tertentu, kehadiran program atau file tertentu, atau kapasitas disk yang melewati batas. Tidak semua virus mempunyai tahap ini.
     - ⇒ Fase propagasi (propagation phase). Virus menempatkan kopian dirinya ke program lain atau daerah sistem tertentu di disk. Program yang terinfeksi virus akan mempunyai kloning virus. Kloning virus itu dapat kembali memasuki fase propagasi.
     - ⇒ Fase pemicuan (triggering phase). Virus diaktifkan untuk melakukan fungsi tertentu. Seperti pada fase tidur, fase pemicuan dapat disebabkan beragam kejadian sistem termasuk penghitungan jumlah kopian dirinya.
     - ⇒ Fase eksekusi (execution phase). Virus menjalankan fungsinya, fungsinya mungkin sepele seperti sekedar menampilkan pesan dilayar atau merusak seperti merusak program dan file-file data, dan sebagainya. Kebanyakan virus melakukan kerjanya untuk suatu sistem operasi tertentu, lebih spesifik lagi pada platform perangkat keras tertentu. Virus-virus dirancang memanfaatkan rincian-rincian dan kelemahan-kelemahan sistem tertentu.

   - **Klasifikasi tipe virus** :
     a. Parasitic virus. Merupakan virus tradisional dan bentuk virus yang paling sering. Tipe ini mencantolkan dirinya ke file .exe. Virus mereplikasi ketika program terinfeksi dieksekusi dengan mencari file-file .exe lain untuk diinfeksi.
     b. Memory resident virus. Virus memuatkan diri ke memori utama sebagai bagian program yang menetap. Virus menginfeksi setiap program yang dieksekusi.
     c. Boot sector virus. Virus menginfeksi master boot record atau boot record dan menyebar saat sistem diboot dari disk yang berisi virus.

d. Stealth virus. Virus yang bentuknya telah dirancang agar dapat menyembunyikan diri dari deteksi perangkat lunak antivirus.
e. Polymorphic virus. Virus bermutasi setiap kali melakukan infeksi. Deteksi dengan penandaan virus tersebut tidak dimungkinkan. Penulis virus dapat melengkapi dengan alat-alat bantu penciptaan virus baru (virus creation toolkit, yaitu rutin-rutin untuk menciptakan virus-virus baru). Dengan alat bantu ini penciptaan virus baru dapat dilakukan dengan cepat. Virus-virus yang diciptakan dengan alat bantu biasanya kurang canggih dibanding virus-virus yang dirancang dari awal.

6. **Worm** : Program yang dapat mereplikasi dirinya dan mengirim kopian-kopian dari komputer ke komputer lewat hubungan jaringan. Begitu tiba, worm diaktifkan untuk mereplikasi dan progasai kembali. Selain hanya propagasi, worm biasanya melakukan fungsi yang tak diinginkan.
   - Network worm menggunakan hubungan jaringan untuk menyebar dari sistem ke sistem lain. Sekali aktif di suatu sistem, network worm dapat berlaku seperti virus atau bacteria, atau menempelkan program trojan horse atau melakukan sejumlah aksi menjengkelkan atau menghancurkan.
   - Untuk mereplikasi dirinya, network worm menggunakan suatu layanan jaringan, seperti : Fasilitas surat elektronik (electronic mail facility), yaitu worm mengirimkan kopian dirinya ke sistem-sistem lain.
   - Kemampuan eksekusi jarak jauh (remote execution capability), yaitu worm mengeksekusi kopian dirinya di sistem lain.
   - Kemampuan login jarak jauh (remote login capability), yaitu worm log pada sistem jauh sebagai pemakai dan kemudian menggunakan perintah untuk mengkopi dirinya dari satu sistem ke sistem lain. Kopian program worm yang baru kemudian dijalankan di sistem jauh dan melakukan fungsi-fungsi lain yang dilakukan di sistem itu, worm terus menyebar dengan cara yang sama.
   - Network worm mempunyai ciri-ciri yang sama dengan virus komputer, yaitu mempunyai fase-fase sama, yaitu : Dormant phase, Propagation phase, Trigerring phase, Execution phase.
   - Network worm juga berusaha menentukan apakah sistem sebelumnya telah diinfeksi sebelum mengirim kopian dirinya ke sistem itu.

Antivirus

Solusi ideal terhadap ancaman virus adalah **pencegahan**. Jaringan diijinkan virus masuk ke sistem. Sasaran ini, tak mungkin dilaksanakan sepenuhnya. Pencegahan dapat mereduksi sejumlah serangan virus. Setelah pencegahan terhadap masuknya virus, maka pendekatan berikutnya yang dapat dilakukan adalah :

1. **Deteksi**. Begitu infeksi telah terjadi, tentukan apakah infeksi memang telah terjadi dan cari lokasi virus.
2. **Identifikasi**. Begitu virus terdeteksi maka identifikasi virus yang menginfeksi program.
3. **Penghilangan**. Begitu virus dapat diidentifikasi maka hilangkan semua jejak virus dari program yang terinfeksi dan program dikembalikan ke semua (sebelum terinfeksi). Jika deteksi virus sukses dilakukan, tapi identifikasi atau penghilangan jejak tidak dapat dilakukan, maka alternatif yang dilakukan adalah menghapus program yang terinfeksi dan kopi kembali backup program yang masih bersih.

Perkembangan program antivirus dapat diperiode menjadi empat generasi, yaitu :

1. **Generasi pertama :** sekedar scanner sederhana. Antivirus menscan program untuk menemukan penanda (signature) virus. Walaupun virus mungkin berisi karakter-karakter varian, tapi secara esensi mempunyai struktur dan pola bit yang sama di semua kopiannya. Teknis ini terbatas untuk deteksi virus-virus yang telah dikenal. Tipe lain antivirus generasi pertama adalah mengelola rekaman panjang (ukuran) program dan memeriksa perubahan panjang program.

2. **Generasi kedua :** scanner yang pintar (heuristic scanner). Antivirus menscan tidak bergantung pada penanda spesifik. Antivirus menggunakan aturan-aturan pintar (heuristic rules) untuk mencari kemungkinan infeksi virus.Teknik yang dipakai misalnya mencari fragmen- fragmen kode yang sering merupakan bagian virus. Contohnya, antivirus mencari awal loop enkripsi yang digunakan polymorphic virus dan menemukan kunci enkripsi. Begitu kunci ditemukan, antivirus dapat mendeskripsi virus untuk identifikasi dan kemudian menghilangkan infeksi virus. Teknik ini adalah pemeriksanaan integritas. Checksum dapat ditambahkan di tiap program. Jika virus menginfeksi program tanpa mengubah checksum, maka pemeriksaan integritas akan menemukan perubahan itu. Untuk menanggulangi virus canggih yang mampu mengubah checksum saat menginfeksi program, fungsi hash terenkripsi digunakan. Kunci enkripsi disimpan secara terpisah dari program sehingga program tidak dapat menghasilkan kode hash baru dan mengenkripsinya. Dengan menggunakan fungsi hash bukan checksum sederhana maka mencegah virus menyesuaikan program yang menghasilkan kode hash yang sama seperti sebelumnya.
3. **Generasi ketiga :** jebakan-jebakan aktivitas (activity trap). Program antivirus merupakan program yang menetap di memori (memory resident program). Program ini mengidentifikasi virus melalui aksi- aksinya bukan dari struktur program yang diinfeksi. Dengan antivirus semacam in tak perlu mengembangkan penanda-penanda dan aturan-aturan pintar untuk beragam virus yang sangat banyak. Dengan cara ini yang diperlukan adalah mengidentifikasi kumpulan instruksi yang berjumlah sedikit yang mengidentifikasi adanya usaha infeksi. Kalau muncul kejadian ini, program antivirus segera mengintervensi.
4. **Generasi keempat :** proteksi penuh (full featured protection). Antivirus generasi ini menggunakan beragam teknik antivirus secara bersamaan. Teknik-teknik ini meliputi scanning dan jebakan-jebakan aktivitas. Antivirus juga mempunyai senarai kapabilitas pengaksesan yang membatasi kemampuan virus memasuki sistem dan membatasi kemampuan virus memodifikasi file untuk menginfeksi file. Pertempuran antara penulis virus dan pembuat antivirus masih berlanjut. Walau beragam strategi lebih lengkap telah dibuat untuk menanggulangi virus, penulis virus pun masih berlanjut menulis virus yang dapat melewati barikade-barikade yang dibuat penulis antivirus. Untuk pengaman sistem komputer, sebaiknya pengaksesandan pemakaian komputer diawasi dengan seksama sehingga tidak menjalankan program atau memakai disk yang belum terjamin kebersihannya dari infeksi virus. Pencegahan terbaik terhadap ancaman virus adalah mencegah virus memasuki sistem disaat yang pertama.