Keamanan Sistem Komputer

Unintentional, Malicious Code – Malware, Countermeasures

Introduction

- Program terdiri dari sekumpulan 0 dan 1, merepresentasikan perintah seperti: memindahkan sebuah data, membandingkan data, atau bercabang ke perintah lain.
- Contoh penggunaan program: *satellite control, smart-home technology, digital photography, streaming video, social networks,* dll.
- Masalah keamanan pada program dapat berada dimana pun di antara *hardware* dan *user interface*.
- *Security failures* dapat terjadi karena unsur kesengajaan (*intentional*) atau pun tidak disengaja (*nonmalicious*); namun keduanya menimbulkan kerusakan.

Terminology

- Error

  Kesalahan yang dibuat manusia
- Fault

  Kesalahan command, proses, atau definisi data pada program yang disebabkan oleh error
- Failure

  Dapat ditemukan sebelum atau sesudah system delivery, saat testing, operasi, ataupun maintenance. Fault terlihat dari dalam sistem oleh developer, sedangkan failure terlihat dari luar oleh user.
- Flaw

  Istilah yang digunakan security engineer untuk fault dan failure.

Unintentional (Nonmalicious) Programming Oversights

- Semakin berkembang dan kompleks suatu program, semakin sedikit yang user ketahui tentang cara kerja program.
- User jarang menyadari apakah program yang digunakan memberikan hasil yang benar.
- Flaws pada program memiliki dampak pada keamanan: masalah integrity dan memberikan peluang untuk eksploitasi bagi malicious user.

Unintentional (Nonmalicious) Programming Oversights – cont.

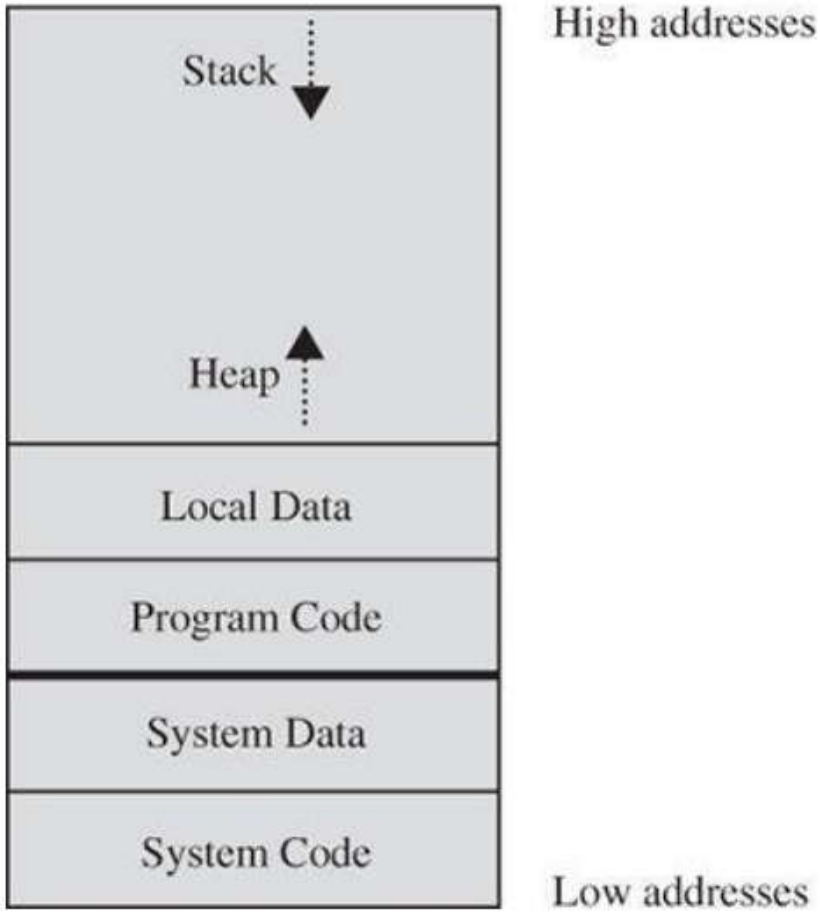
- Flaw pada program dapat mempengaruhi kebenaran dari keluaran program, sehingga mempengaruhi integrity.
- Integrity tidak hanya mencakup kebenaran, namun juga akurasi, presisi, dan konsistensi.
- Jika attacker mengetahui flaw pada program, dapat digunakan untuk memanipulasi perilaku program dan dapat berujung pada serangan.

Buffer Overflow

- Awalnya hanya menjadi gangguan kecil yang menyebabkan error atau bahkan system crash.
- Kemudian digunakan attacker untuk menyebabkan system crash kemudian controlled failure, dan berakibat pada permasalahan keamanan.
- Memori memiliki ukuran terbatas, namun bersifat fleksibel dapat digunakan untuk menyimpan code ataupun data.

Buffer Overflow

- Attacker akan berpikir untuk menyelipkan nilai yang dapat menimbulkan kesalahan atau kerusakan.
- Misalnya jika overflow terjadi pada sistem, dapat tejadi pengalihan instruksi yang dieksekusi atau bahkan berhenti dari prosedur yang dilakukan.



Buffer Overflow

Beberapa cara yang dapat dilakukan attacker agar overflow yang terjadi memberikan dampak:

- Overwrite program counter

  Ketika routine berhenti, instruksi berikutnya beralih ke alamat yang sudah dimodifikasi

- Overwrite sebagian dari code di low memory

  Mengganti instruksi yang dilaksanakan

- Overwrite program counter dan data

  Dilakukan pada stack, menyebabkan eksekusi data yang ditulis pada stack.

Buffer Overflow – Countermeasures

Beberapa cara untuk mengatasi overwrite pada memori adalah dengan berada dalam batasan yang ditentukan.

- Cek panjang data sebelum proses writing
- Melakukan pengecekan indeks array masih di dalam batasan
- Double-check pada condition code untuk mengetahui jika terjadi error
- Mengawasi input dan hanya menerima karakter sebanyak yang dapat diproses
- Membatasi hak program

Incomplete Mediation

- Mediation, proses verifikasi apakah subjek memiliki hak untuk melakukan operasi pada suatu objek.
- Incomplete mediation dapat digunakan oleh attacker untuk menyebabkan masalah keamanan.

Incomplete Mediation

```
http://www.somesite.com/subpage/userinput.asp?
parm1=(808)555-1212&parm2=2015Jan17
```

- parm1 -> nomor telepon?
- parm2 -> tanggal?
- Bagaimana jika parm2 diubah menjadi 1800Jan01? 1800Feb30? 2048Min32?
- Kesalahan yang terjadi karena user, seharusnya tidak menyebabkan program failures.
- Validasi input harus dilakukan dengan cara membatasi cara memasukkan data.

Incomplete Mediation – Countermeasures

- Solusinya adalah complete mediation.
- Caranya adalah dengan menggunakan access control yang diimplementasikan berdasarkan konsep reference monitor.
- Karakteristik dari reference monitor:
  1. Simple
  2. Unbypassable
  3. Always invoked

Time-of-Check to Time-of-Use

- Antara waktu pengecekan dan penggunaan, data harus dilindungi dari hal-hal yang dapat mengakibatkan perubahan.
- Dampak dari segi keamanan: ineffective access control, confidentiality failure, integrity failure.

Time-of-Check to Time-of-Use – Countermeasures

- Menggunakan access control, access-checking software harus memiliki data yang akan diproses sampai setelah proses selesai dilakukan.
- Cara lain dengan memastikan serial integrity. Tidak boleh ada interupsi saat proses validasi berlangsung. Proses validasi dapat menyalin data dari user ke area yang tidak terjangkau oleh user, melakukan proses validasi, dan menyegel hasil validasi untuk mendeteksi modifikasi.

Undocumented Access Point

- Pada saat development, programmer bisa saja membuat entry point ataupun execution mode yang tidak terdokumentasi.
- Terkadang programmer lupa untuk menghilangkan entry point tersebut, atau dengan sengaja membiarkannya untuk proses maintenance selanjutnya dengan pemikiran bahwa tidak ada orang lain yang akan menemukannya.

Undocumented Access Point

- Access point yang tidak terdokumentasi disebut backdoor atau trapdoor.
- Secret backdoor pada akhirnya dapat ditemukan dan dieksploitasi oleh orang lain.

Off-by-One Error

- Kesalahan kalkulasi pada proses penghentian proses perulangan, atau tidak memperhatikan jumlah elemen pada array.

- Cara mengatasinya? Correct programming. Selalu memastikan bahwa tempat penyimpanan data cukup besar untuk menampung data.

Integer Overflow

- Terjadi karena lokasi penyimpanan memiliki ukuran terbatas yang hanya dapat menyimpan integer hingga batas tertentu.

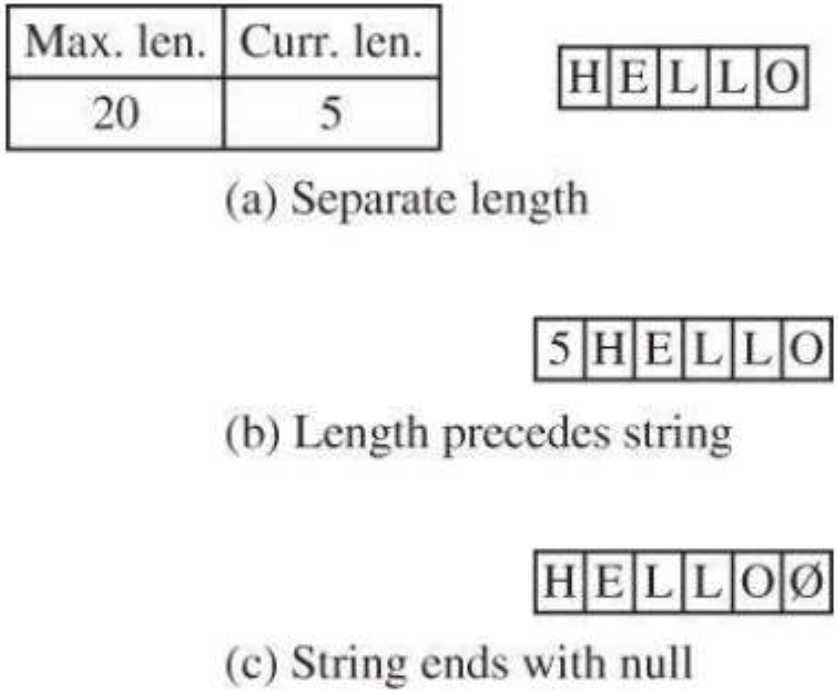
Word Size	Signed Values	Unsigned Values
8 bits	−128 to +127	0 to 255 ($2^8 - 1$)
16 bits	−32,768 to +32,767	0 to 65,535 ($2^{16} - 1$)
32 bits	−2,147,483,648 to +2,147,483,647	0 to 4,294,967,296 ($2^{32} - 1$)

Unterminated Null-Terminated String

- Seringkali buffer overflow terjadi karena attacker secara sengaja memberikan string yang terlalu panjang kepada program untuk melihat apakah program akan mengalami failure.
- Terkadang terjadi secara tidak sengaja, dimana program salah menulis lebih banyak part dari string, sehingga string memiliki ukuran lebih panjang dari seharusnya.

Unterminated Null-Terminated String

- Karakter dengan panjang bervariasi dibatasi dengan tiga cara.
- Cara ketiga (c) disebut null terminated, berarti string diakhiri dengan byte null 0x00.
- Buffer overflow dapat terjadi pada cara (c), jika terjadi kesalahan pada penulisan byte terakhir (terminating null character), maka aplikasi akan terus membaca data pada memori melebihi ukuran string hingga ditemukan null byte.



Parameter Length, Type, and Number

Disebabkan oleh:

- Terlalu banyak parameter
- Kesalahan tipe atau ukuran output
- String terlalu panjang

Race Condition

- Perilaku program bergantung pada urutan eksekusi dari dua prosedur yang “berkompetisi” (misalnya untuk mendapatkan resource)
- Race condition bergantung pada urutan dan timing dari dua proses berbeda, membuatnya susah untuk ditemukan (dan diuji)

Malicious Code – Malware

- Program atau bagian program yang ditanam oleh pihak dengan malicious intent untuk menyebabkan efek yang tidak diharapkan.
- Dapat disebabkan oleh worm atau virus.

Malware – Viruses, Trojan Horses, Worms

- Virus adalah program yang dapat menggandakan dirinya dan memberikan malicious code pada nonmalicious program dengan memodifikasi program tersebut.
- Worm adalah program yang menyebarkan replica dirinya melalui jaringan.
- Worm beroperasi melalui jaringan, sedangkan virus dapat menggunakan media apapun (namun paling banyak melalui salinan program atau file). Worm menyebarkan replica dirinya sebagai stand alone program, sedangkan virus menumpang pada program lain.
- Trojan horse adalah malicious code yang masuk ke dalam program tanpa terdeteksi dan menimbulkan malicious effect setelahnya.

Malicious Code

- Malicious code muncul pada 1970-an, dengan perkembangan yang sangat pesat.
- Awalnya tersebar dari perorangan melalui media (contoh removable disk) atau dokumen yang dikirimkan melalui email.
- Setelah pengguna internet bertambah pesat, pertumbuhan persebaran malicious code juga semakin pesat.
- Zero-Day Attack: Malware mengeksploitasi kelemahan produk dimana pemilik produk tidak memiliki countermeasure.
- Malware tidak hanya menyerang user individu tetapi juga industri.

Malicious Code

Dampak dari malicious code:

- Nondestructive

  Memunculkan pesan atau gambar pada layar; bertujuan untuk menimbulkan kepanikan pada user.

- Destructive

  Merusak atau bahkan menghapus file, merusak software, atau menjalankan perintah yang menyebabkan kerusakan.

- Commercial or criminal intent

  Mencoba untuk mengambil alih computer user, memasang code yang dapat mempermudah proses pengumpulan data personal.

Malicious Code

Harms to Users:

- Menampilkan tulisan atau gambar pada layar
- Membuka jendela browser yang berkaitan dengan aktivitas saat itu (contoh: membuka situs maskapai penerbangan saat sedang membuka situs destinasi luar negeri)
- Mengirimkan email ke sebagian atau seluruh kontak
- Menghapus semua file
- Mengubah system program file
- Mengubah system information, seperti Windows registry

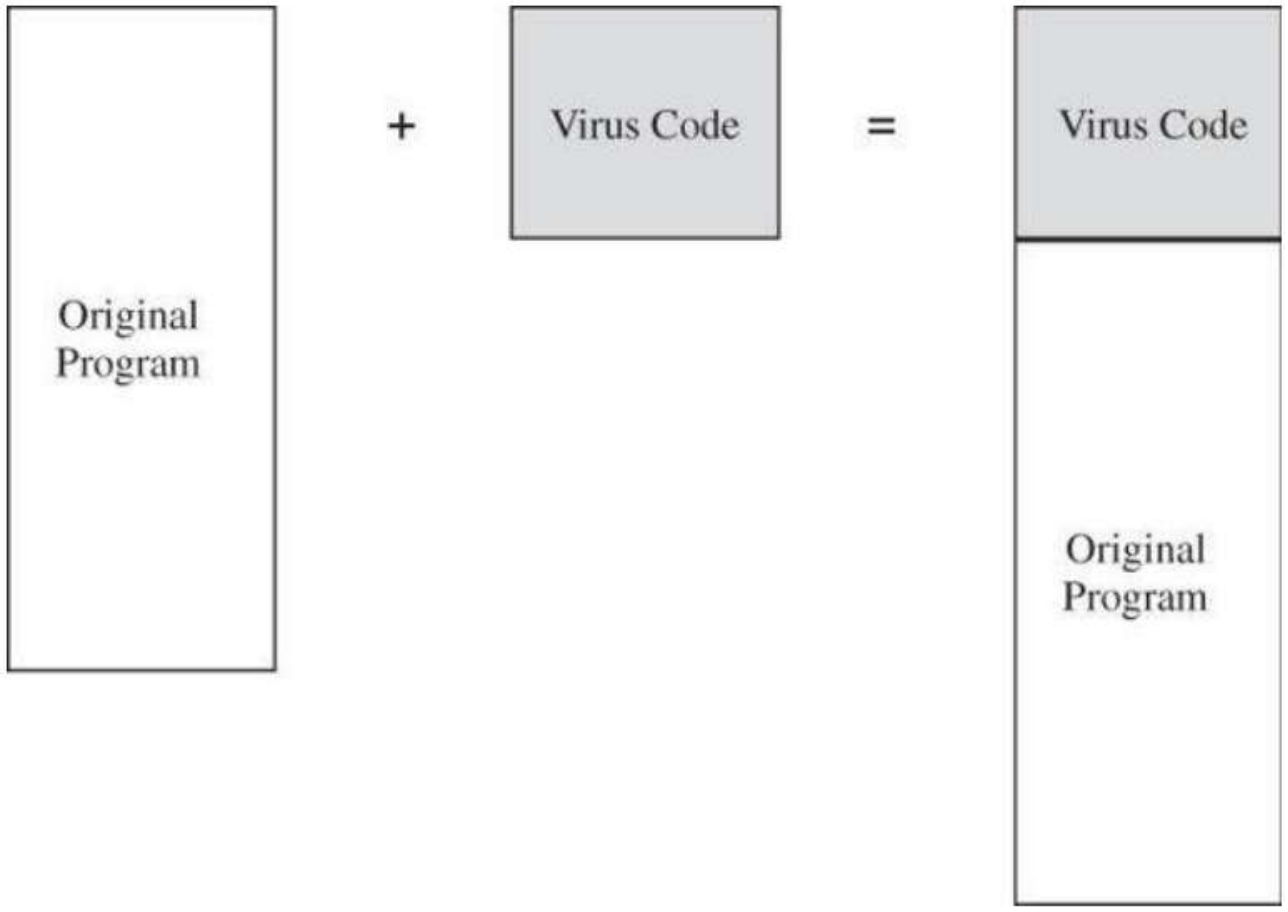
Malicious Code

- Untuk melakukan tugasnya dan menyebarkan dirinya, malware harus dieksekusi.
- Ada banyak cara untuk memastikan program dieksekusi pada computer yang aktif:
  - Transmisi dari file setup dan installer
  - Attached pada file dan tereksekusi ketika file dibuka
  - Document viruses
  - Autorun
  - Distribusi dari flash memory

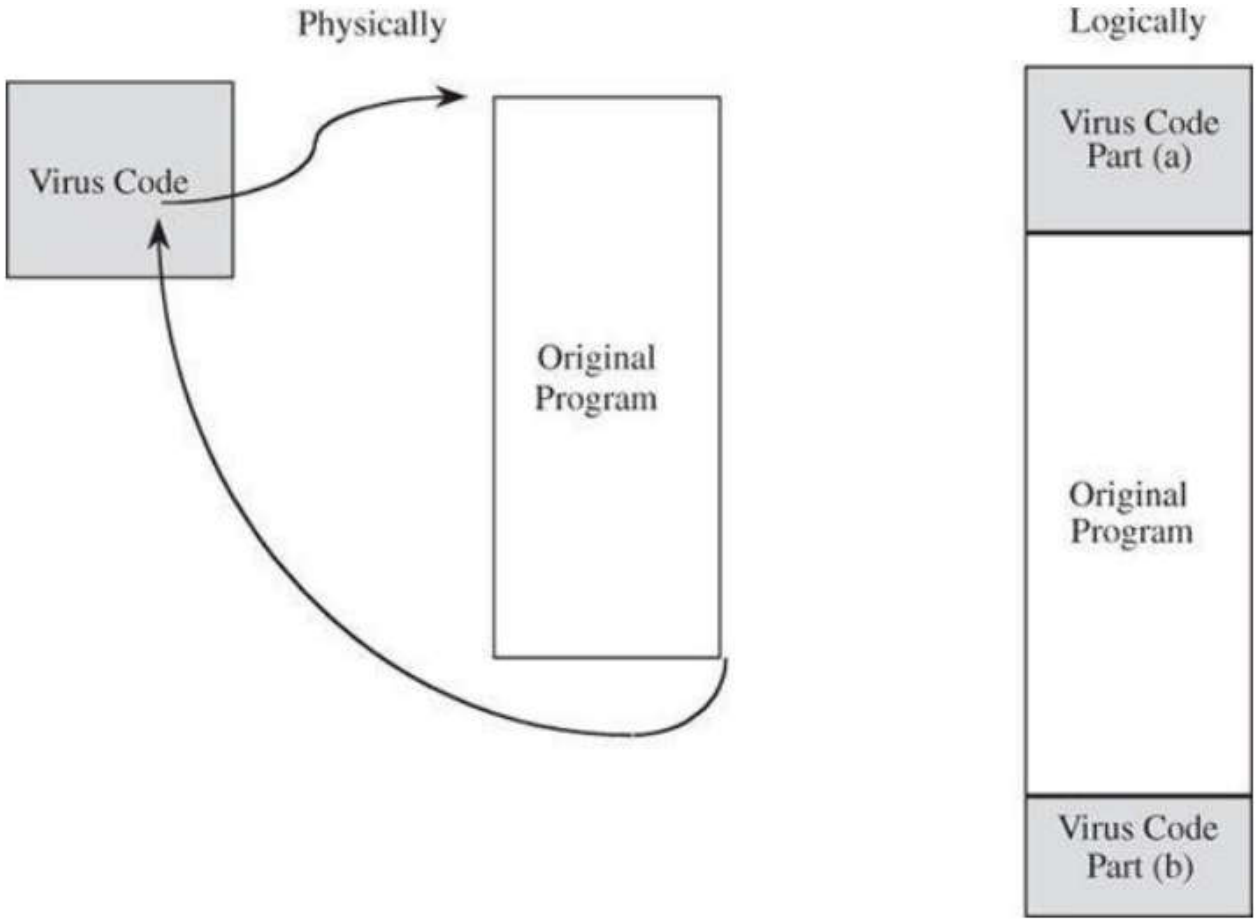
Virus Attachment

- Appended Virus
- Surround a Program
- Integrated Virus and Replacement

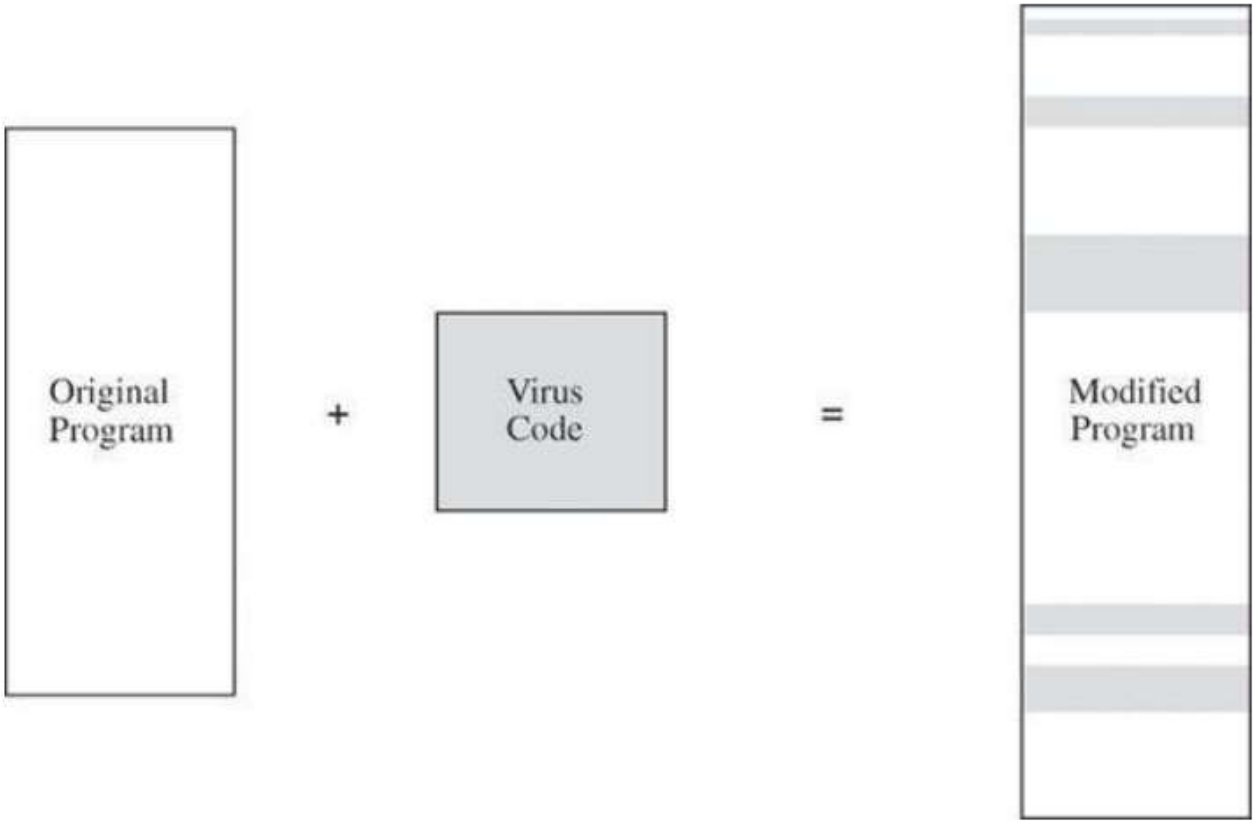
Appended Virus



Surround a Program



Integrated Virus and Replacements



Malicious Code

- Kebanyakan virus menjaga dirinya dengan menyembunyikan aksinya, tidak memperlihatkan keberadaannya, dan menyamarkan dirinya.
- Salah satu caranya adalah dengan menyembunyikan code pada file dengan data yang besar, seperti gambar atau read-only documents, dengan proses steganografi.
- Keberadaan malware toolkits memungkinkan attacker melakukan serangan hanya dengan menekan tombol.

Countermeasures

- Malware berkembang dengan sangat pesat.
- Countermeasure yang tersedia tidak sempurna, sebagian bersifat reaktif (bukan preventif).
- Harus dilakukan oleh semua pihak dari developer hingga user.

Countermeasures for Users

- Menggunakan software dari vendor yang reliable.
- Jika harus menggunakan software dari sumber lain, uji sotware tersebut pada isolated computer (komputer yang tidak terhubung ke jaringan dan tidak memiliki data penting).
- Hanya membuka attachment (dan file lain yang berpotensi terinfeksi) jika benar-benar yakin akan keamanannya.
- Hanya menginstall software jika yakin dengan keamanannya.

Countermeasures for Users

- Mengenali situs yang dapat membahayakan.
- Buat recoverable system image, ketika sistem terinfeksi dapat digunakan untuk reboot dengan aman.
- Lakukan backup pada executable system files. Ketika terjadi infeksi, file yang terinfeksi dapat dihilangkan dan lakukan instalasi ulang dengan backup copies.

Countermeasures for Developers

- Modularity

Membuat code dalam unit kecil yang disebut modules. Setiap bagian dibangun dengan memperhatikan persyaratan: single-purpose, small, simple, independent.

Simplisitas pada pembuatan sotware meningkatkan maintainability.

Countermeasures for Developers

- Encapsulation

Enkapsulasi menyembunyikan detail implementasi dari komponen, tapi tidak selalu berarti komponen tersebut terisolasi.

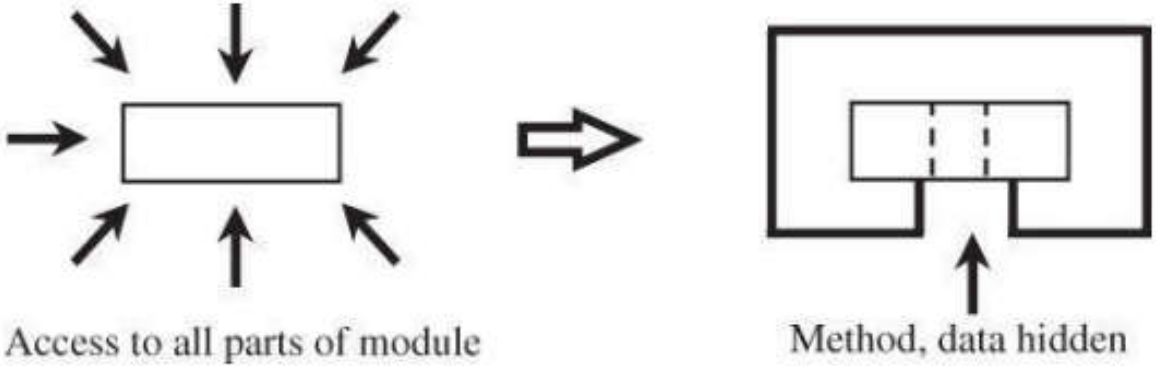
Ketika komponen satu dan lainnya perlu berbagi informasi, proses sharing dilakukan secara hati-hati hanya diketahui oleh komponen yang bersangkutan.

Countermeasures for Developers

- Information Hiding

Mendeskripsikan apa yang dilakukan module, bukan bagaimana module bekerja.

Malicious developer tidak dapat dengan mudah melakukan perubahan pada komponen jika mereka tidak tahu cara kerja komponen.



Countermeasure Specifically for Security

Beberapa asas pembuatan design yang dapat meningkatkan security:

- Least privilege. Setiap user dan program beroperasi menggunakan sedikit mungkin hak akses.
- Open design. Sistem proteksi memiliki mekanisme yang bersifat publik.
- Complete mediation. Setiap akses yang dilakukan harus diperiksa, dimana mekanisme pengecekan tidak bisa dihindari.
- Permission based. Kondisi default adalah penolakan akses. Designer yang konservatif mengidentifikasi apa yang boleh diakses, bukan sebaliknya.

Countermeasure Specifically for Security – cont.

- Separation of privilege. Idealnya, akses pada suatu objek bergantung pada lebih dari satu kondisi, misalnya autentikasi user dan kunci kriptografi.
- Least common mechanism. Menggunakan sistem yang menyediakan pemisahan secara physical maupun logical untuk mengurangi risiko dari proses sharing.
- Easy of use. Jika mekanisme perlindungan mudah digunakan, kecil kemungkinan untuk dihindari.

Penetration Testing for Security

- Penetration testing (ethical hacking) masuk ke dalam sistem, mengidentifikasi bahkan mengeksploitasi kelemahan sistem dengan tujuan untuk pengujian sistem.
- Suatu sistem yang gagal dalam penetration test, pasti memiliki faults.
- Tetapi sistem yang lolos dalam penetration test belum tentu fault-free. Sistem tersebut bisa saja hanya bebas dari fault yang diuji pada proses testing.

Countermeasures that Don’t Work

- Penetrate-and-patch

  Gagal karena harus dilakukan secara cepat, terfokus pada failure tertentu tidak pada sistem keseluruhan.

- Security by Obscurity

  Yakin bahwa sistem akan aman selama tidak ada orang lain di luar implementation group yang mengetahui mekanisme di dalam sistem. Namun sesuatu yang seharusnya tersembunyi selalu dapat ditemukan. Attacker dapat menemukan dan mengeksploitasi banyak hal.

- A Perfect Good-Bad Code Separator

  Apakah program bisa membedakan good program dari bad program? Tidak.

Conclusion

- Error yang disebabkan oleh kesalahan programmer dan kelemahan yang dieksploitasi attacker dapat menimbulkan efek yang serius.
- Dengan menggunakan countermeasures kita dapat mengurangi dampak yang ditimbulkan.