A Very Short Introduction to Software Security

Budi Rahardjo

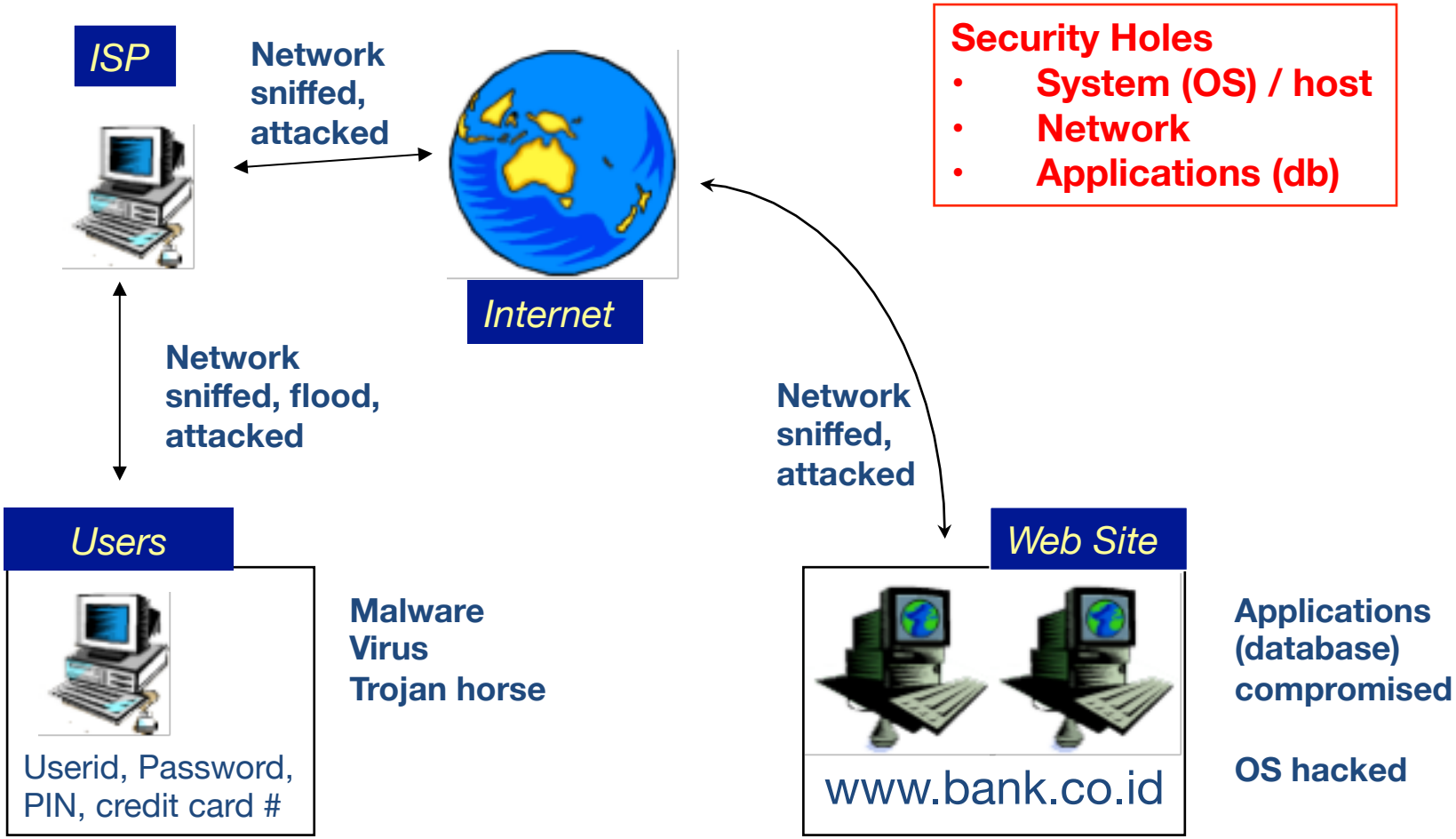
@rahard

2017

Security

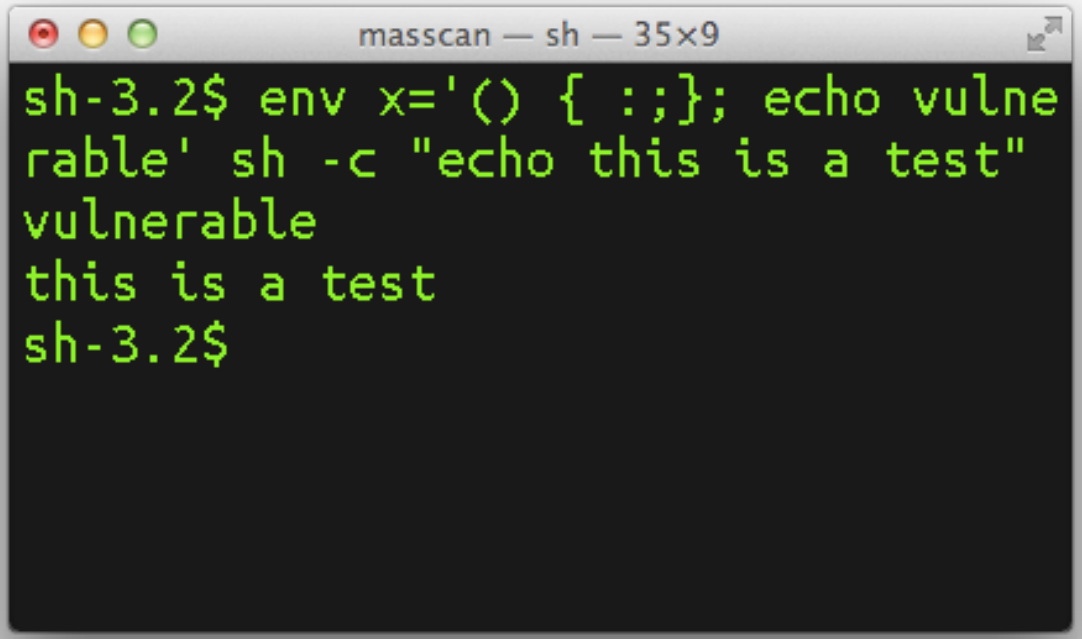
- Network security
- Host / computer security
- Application security

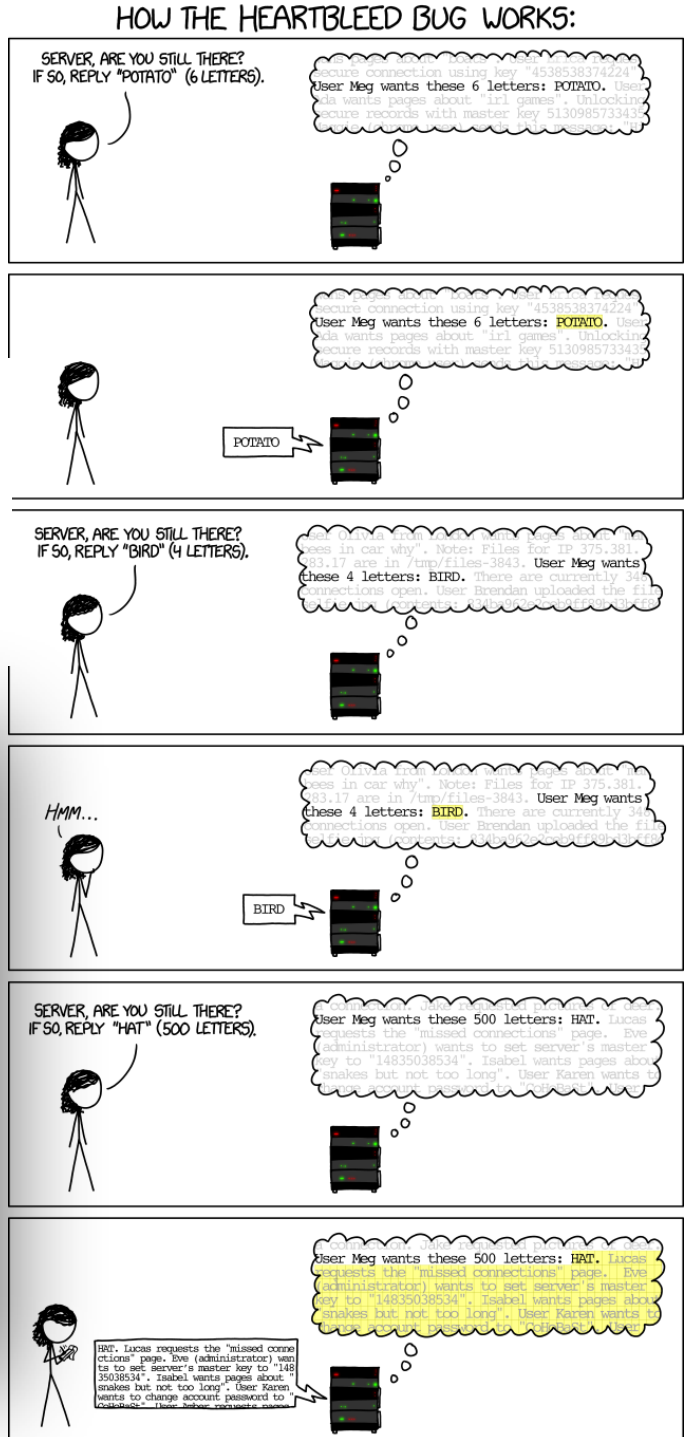
Potensi Lubang Keamanan



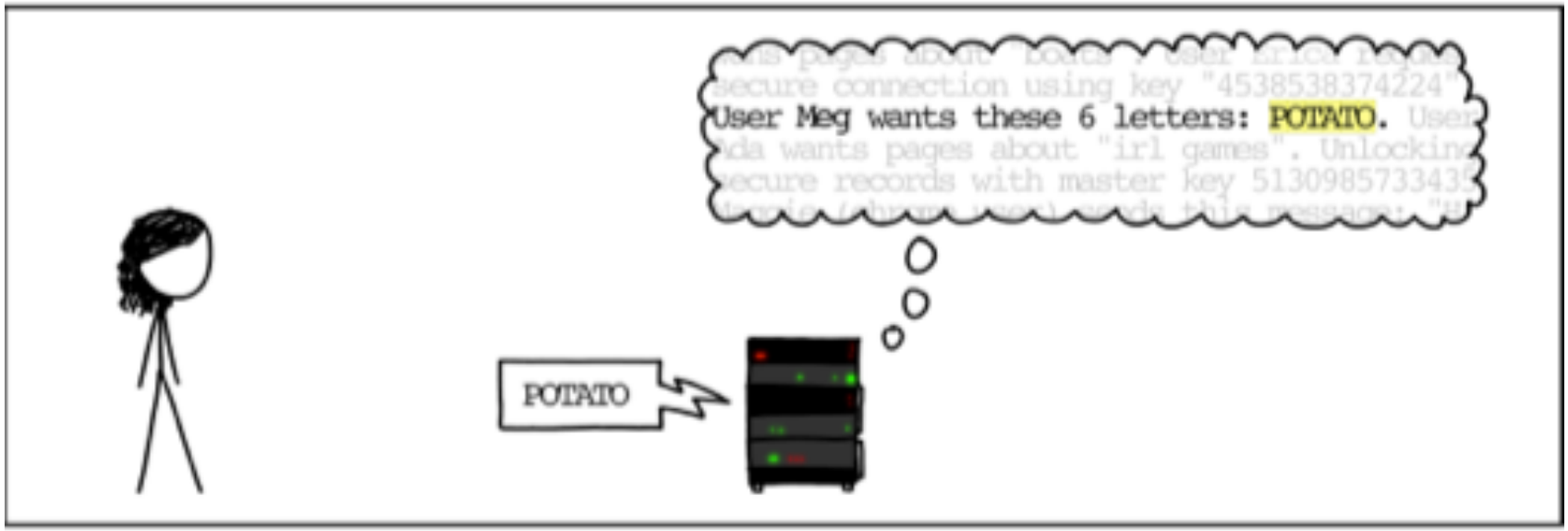
2014 Issues

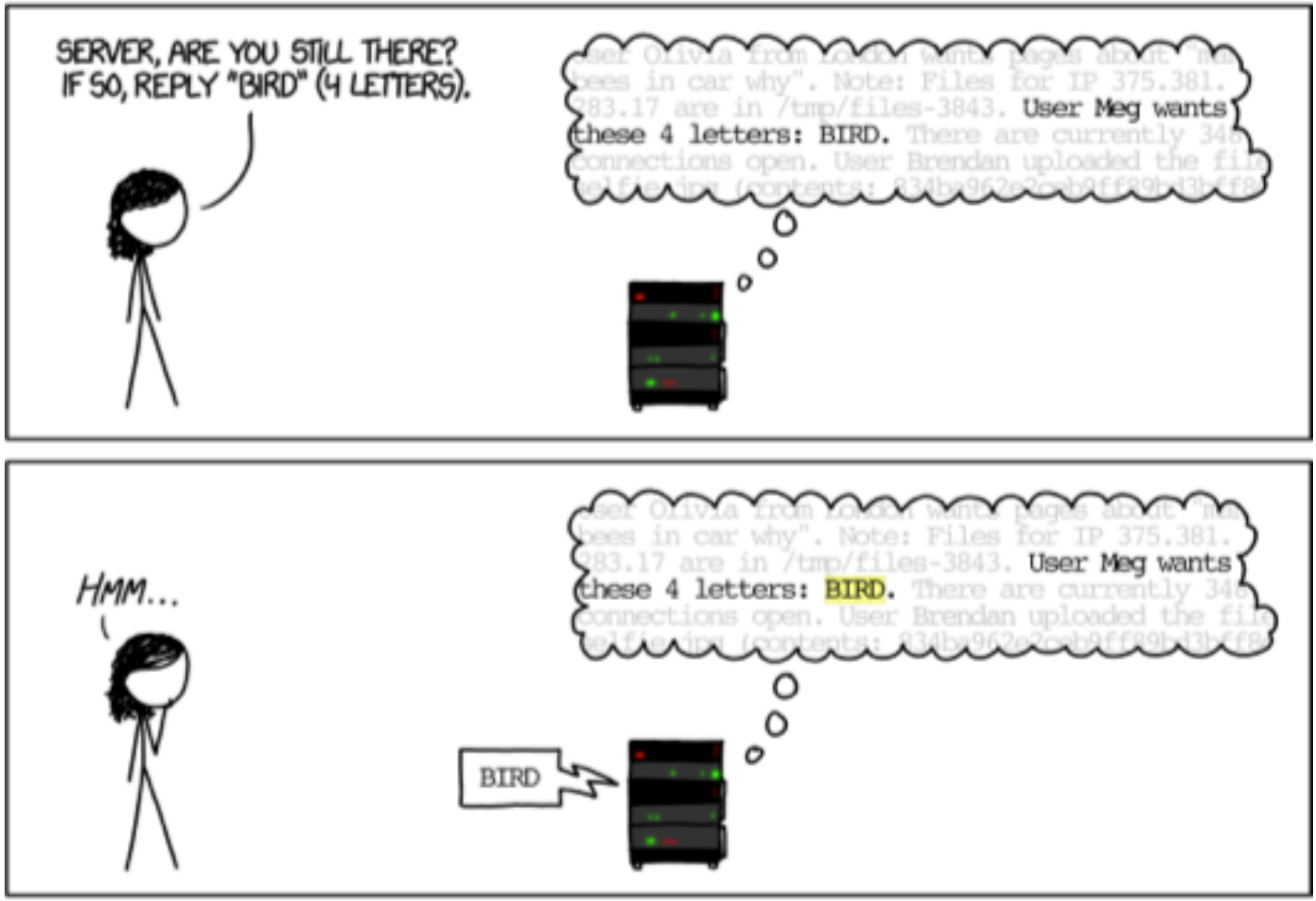
- Bashbug
- Heartbleed

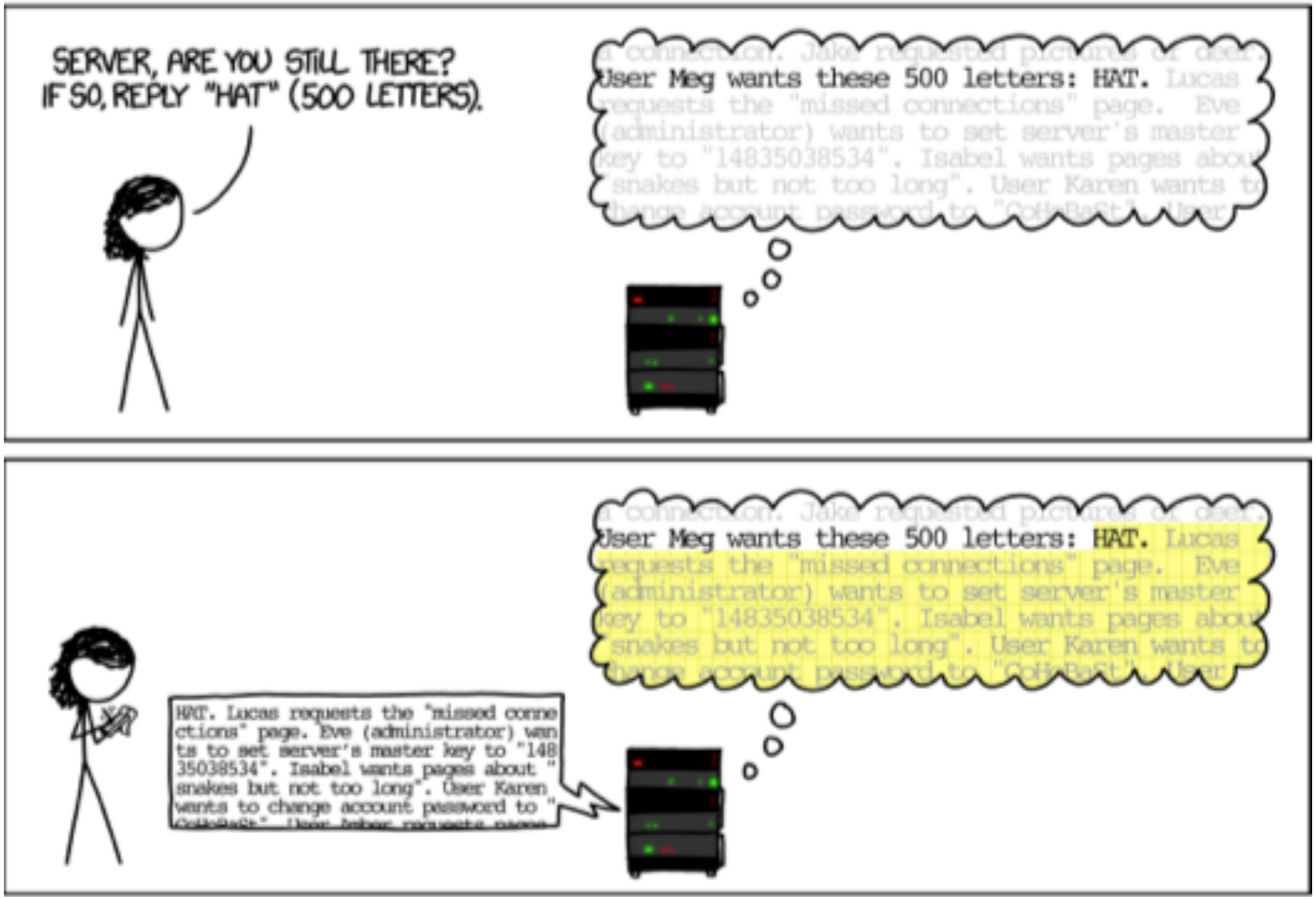




SSL Heartbeat


secure connection using key "4538538374224"
User Meg wants these 6 letters: POTATO.
wants pages about "irl games". Unlocking
secure records with master key 5130985733435
POTATO


SERVER, ARE YOU STILL THERE? IF SO, REPLY "BIRD" (4 LETTERS).
User Meg wants these 4 letters: BIRD.
HMM...
BIRD


SERVER, ARE YOU STILL THERE? IF SO, REPLY "HAT" (500 LETTERS).
User Meg wants these 500 letters: HAT. Lucas requests the "missed connections" page. Eve (administrator) wants to set server's master key to "14835038534". Isabel wants pages about "snakes but not too long". User Karen wants to
User Meg wants these 500 letters: HAT. Lucas requests the "missed connections" page. Eve (administrator) wants to set server's master key to "14835038534". Isabel wants pages about "snakes but not too long". User Karen wants to
HAT. Lucas requests the "missed connections" page. Eve (administrator) wants to set server's master key to "14835038534". Isabel wants pages about "snakes but not too long". User Karen wants to change account password to "

Bashbug

masscan — sh — 35×9

```
sh-3.2$ env x='() { :;}; echo vulne
rable' sh -c "echo this is a test"
vulnerable
this is a test
sh-3.2$
```

Bursa Singapura Terganggu

2014

Perdagangan di bursa saham Singapura sempat macet karena software ngadat. **Halaman 24**

KOMPAS GRAMEDIA

Mengapa Software Penting?

- Berbagai perangkat / layanan bergantung pada software
  - Perangkat bisnis (ATM, e-commerce)
  - Alat komunikasi
  - Peralatan medis
  - Transportasi modern
  - Peralatan elektronik rumah tangga
  - …

Konsekuensi Kegagalan

- Kerugian tangible & intangible
- Reputasi rusak & kepercayaan customer hilang
- Berakibat fatal bila operasionalnya terganggu
- Kerugian produktivitas

Apple iOS 7 bug

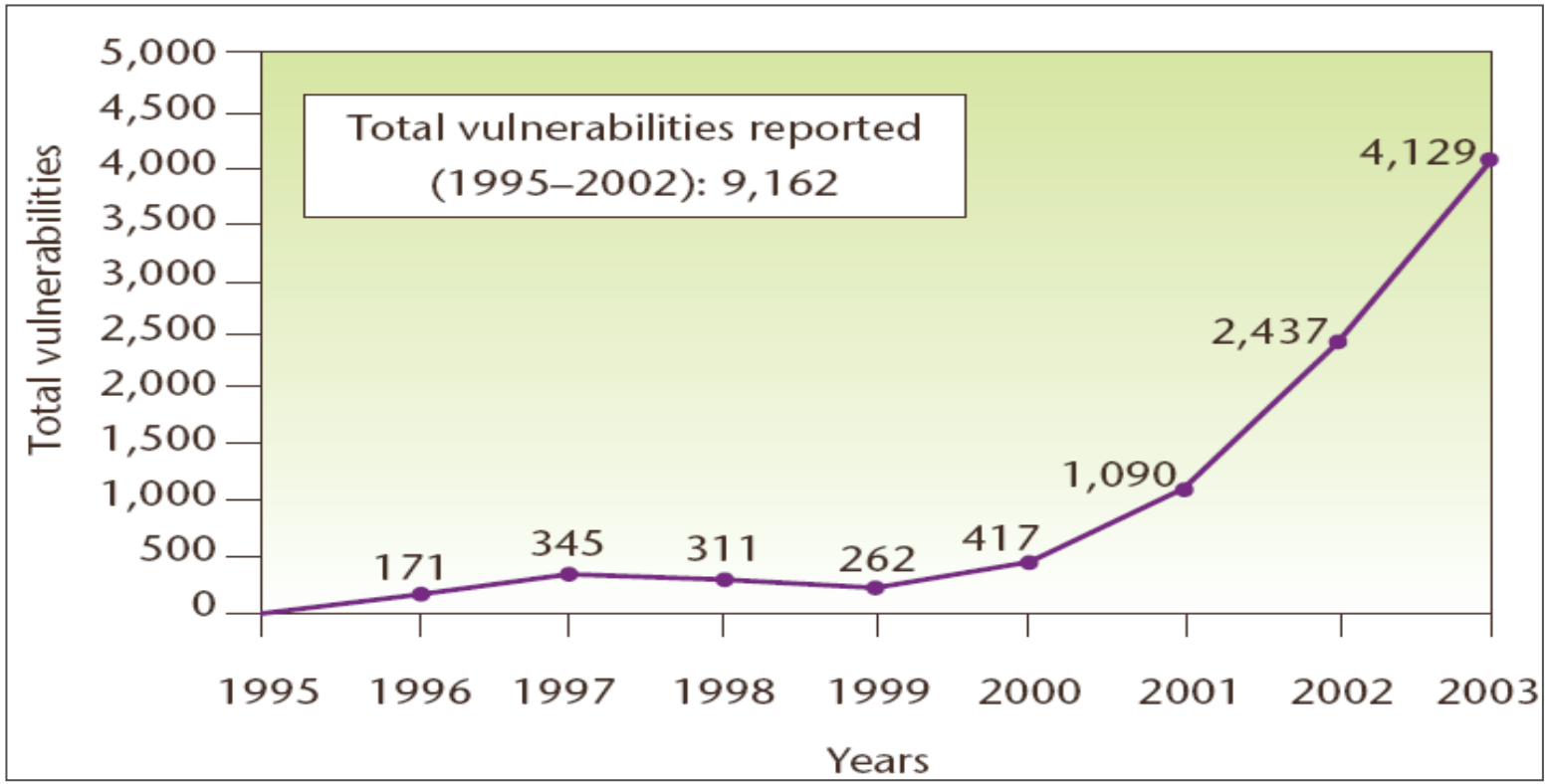
```
static OSStatus
SSLVerifySignedServerKeyExchange(SSLContext *ctx, bool isRsa, SSLBuffer signedParams,
                                 uint8_t *signature, UInt16 signatureLen)
{
        OSStatus        err;
        ...

        if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
                goto fail;
        if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
                goto fail;
                goto fail;
        if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
                goto fail;
        ...

fail:
        SSLFreeBuffer(&signedHashes);
        SSLFreeBuffer(&hashCtx);
        return err;
}
```

Failed to check SSL: goto bug
Must upgrade to iOS 7.0.6

Statistik Software Vulnerabilities



"Processes for Producing Secure Software",
IEEE Security & Privacy, May/June 2004

Sumber Masalah Keamanan Software

- Connectivity
  - Software terhubung ke mana-mana
- Extensibility
  - Software dapat diubah meskipun sudah dipasang dan digunakan (*deployed*)
- Complexity
  - Software semakin kompleks

Connectivity
Software terhubung ke mana-mana

Dahulu

- Software berdiri sendiri (standalone)
- Potensi serangan dilakukan secara fisik saja

Sekarang

- Software terhubung ke Internet (ATM, e-commerce)
  - SCADA bisa diakses melalui Internet
- Potensi serangan datang dari mana-mana

Extensibility

- Software dapat diubah meskipun sudah dipasang dan digunakan (*deployed*)
- Diubah sambil jalan
- Fitur yang belum ada dapat dikembangkan sambil jalan
  - Autoupdate
  - Applet, plug-in, …
  - Trend ke depan (JAVA, .NET) makin bisa dikembangkan
  - Malicious extension bisa masuk ke sistem

Complexity
Software Semakin Kompleks

- Dihitung dari jumlah baris (Lines of Code)
  - Trennya akan makin terus naik
- Semakin banyak jumlah baris:
  - makin meningkat potensi lubang keamanan
  - makin banyak insiden

Operating System	Year	Lines Of Codes
Windows 3.1	1990	3 milion
Windows NT	1996	4 milion
Windows 95	1997	15 milion
Windows NT 4.0	1998	16.5 milion
Windows 98	1999	18 milion
Windows NT 5.0/2K beta	2000	20 milion
Debian GNU/ Linux 2.2	2000	55 milion
Windows 2000	2000	35 milion
Windows XP	2001	40 milion
Red Hat 7.1	2007	50 milion
Windows Vista	2007	50 milion

Pihak Terlibat Dalam Software Security

Developer

beda sudut pandang antara
Developer dan *Security Engineer*:
Build vs. Break

bekerja sama dengan:
Solution Architects &
System Administrators

Pandangan *developer* terhadap *security*:

- asumsi bahwa firewall akan melindungi
- terlalu percaya pada kriptografi
- memeriksa produk setelah jadi

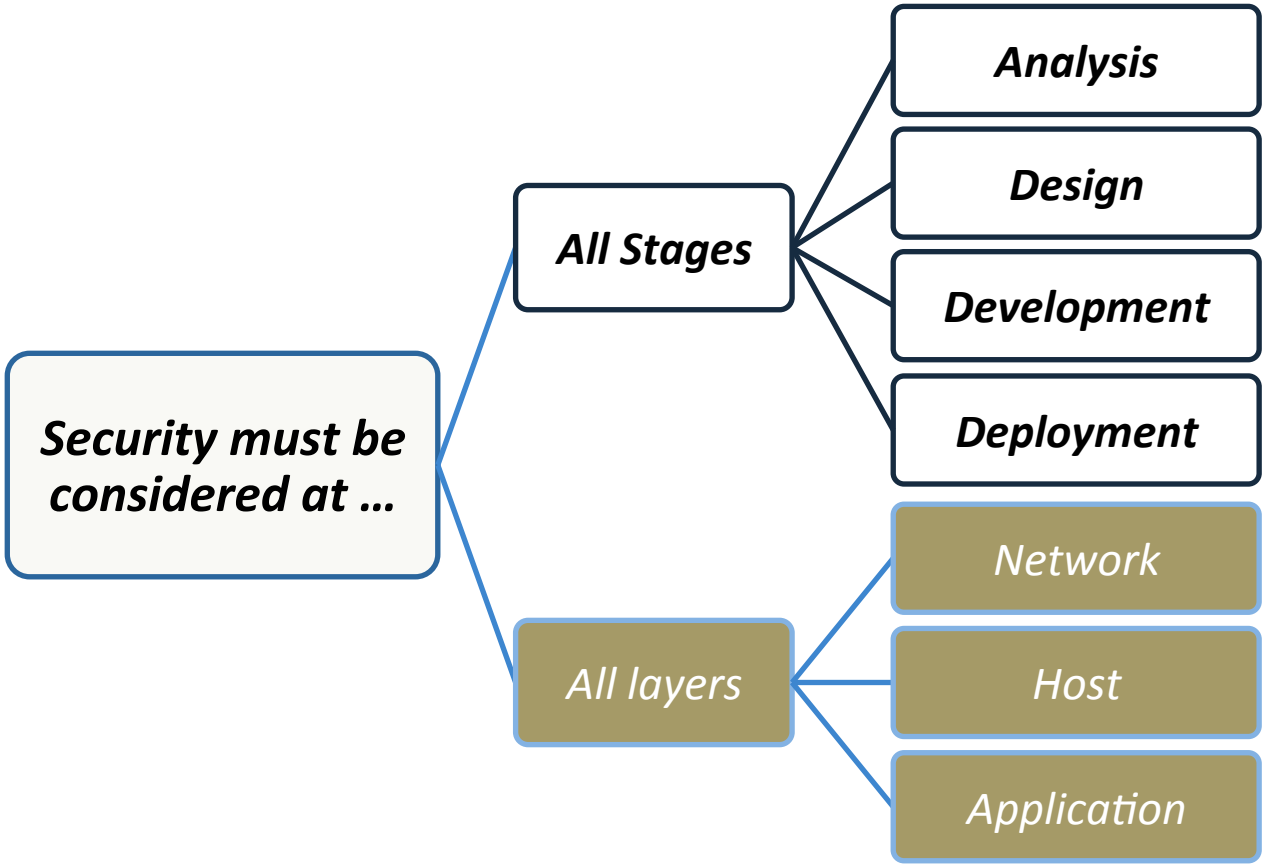
Pandangan *security engineer* terhadap *development*:

- tidak memahami metoda *attack* dan *tool*nya
- tidak bisa melindungi aplikasi dari *attack* secara umum
- selalu mengulang kesalahan *coding*
- terlalu fokus ke spesifikasi

Berkontribusi terhadap *software security* lewat:

- mengadopsi *best-practice* bagi *application security development*
- mengetahui posisi *security vulnerability* dan bagaimana mengatasinya
- menggunakan *secure programming techniques*

Software Security: when & where



Start the Security Process

Security Requirement → ***Software Engineering*** → ***Secure Software***

Membangun *software*/aplikasi yang tetap berperilaku normal ketika ada serangan

Aspek dasar *security*:

- *Confidentiality*
- *Integrity*
- *Availability*

Perbedaan mendasar *Software Safety* dan *Software Security*:

- *Software Security* mempertimbangan keberadaan aktor yang melakukan serangan

Sotware Engineering Evolution

Conventional Software Engineering

- *Design*
- *Design Verification*
- *Coding Testing*

Semua langkah ditujukan untuk menguji bahwa *software* melakukan semua hal yang diinginkan pengembang

Misuse Cases

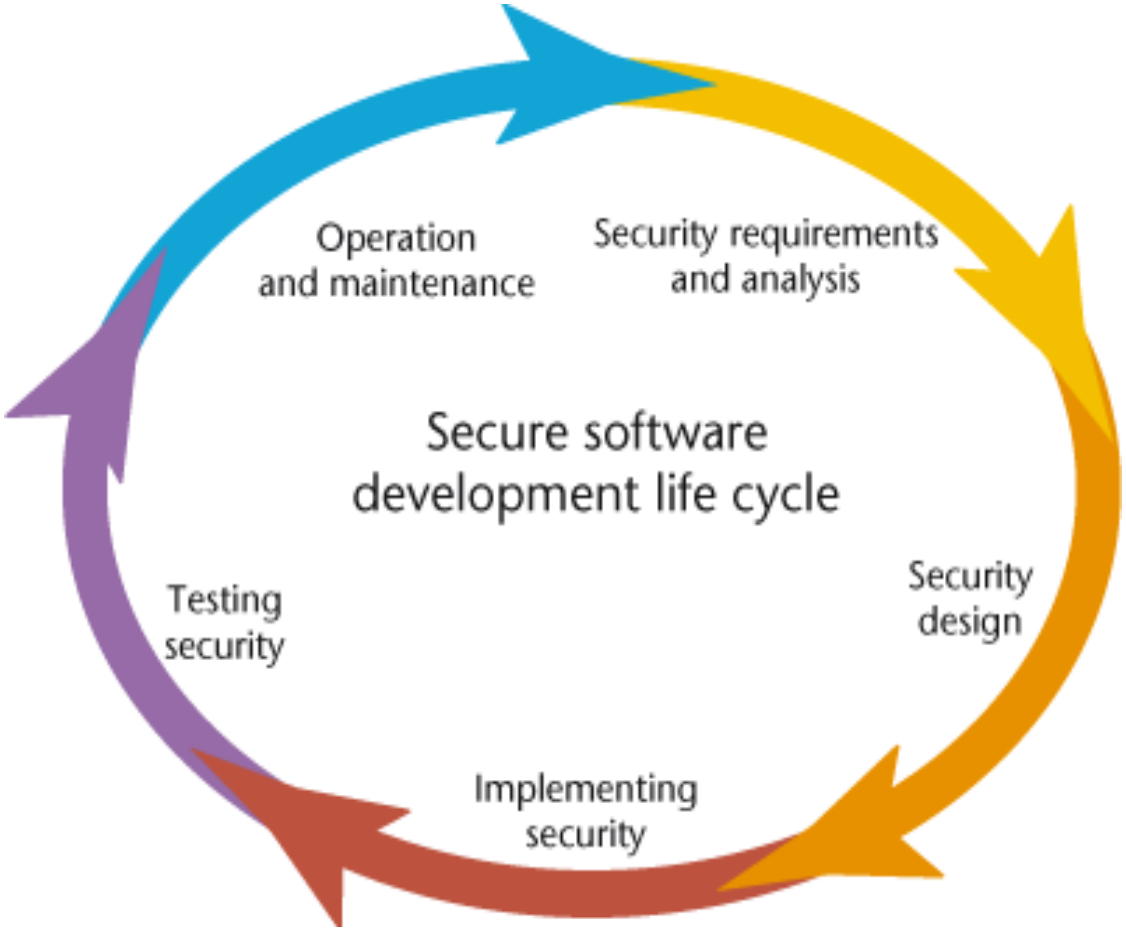
Abuse Cases

Attack Cases

Bagaimana memastikan bahwa *software* tidak melakukan hal-hal lain yang tidak diinginkan???

Bagaimana mengidentifikasi resiko keamanan *software* dan melakukan mitigasi?

Secure Software Development Life Cycle (SSDLC)



Source: www.computer.org

Security Framework: SD^3

Secure by Default

- *secure architecture & code*
- *threat analysis*
- *vulnerability reduction*

Secure by Design

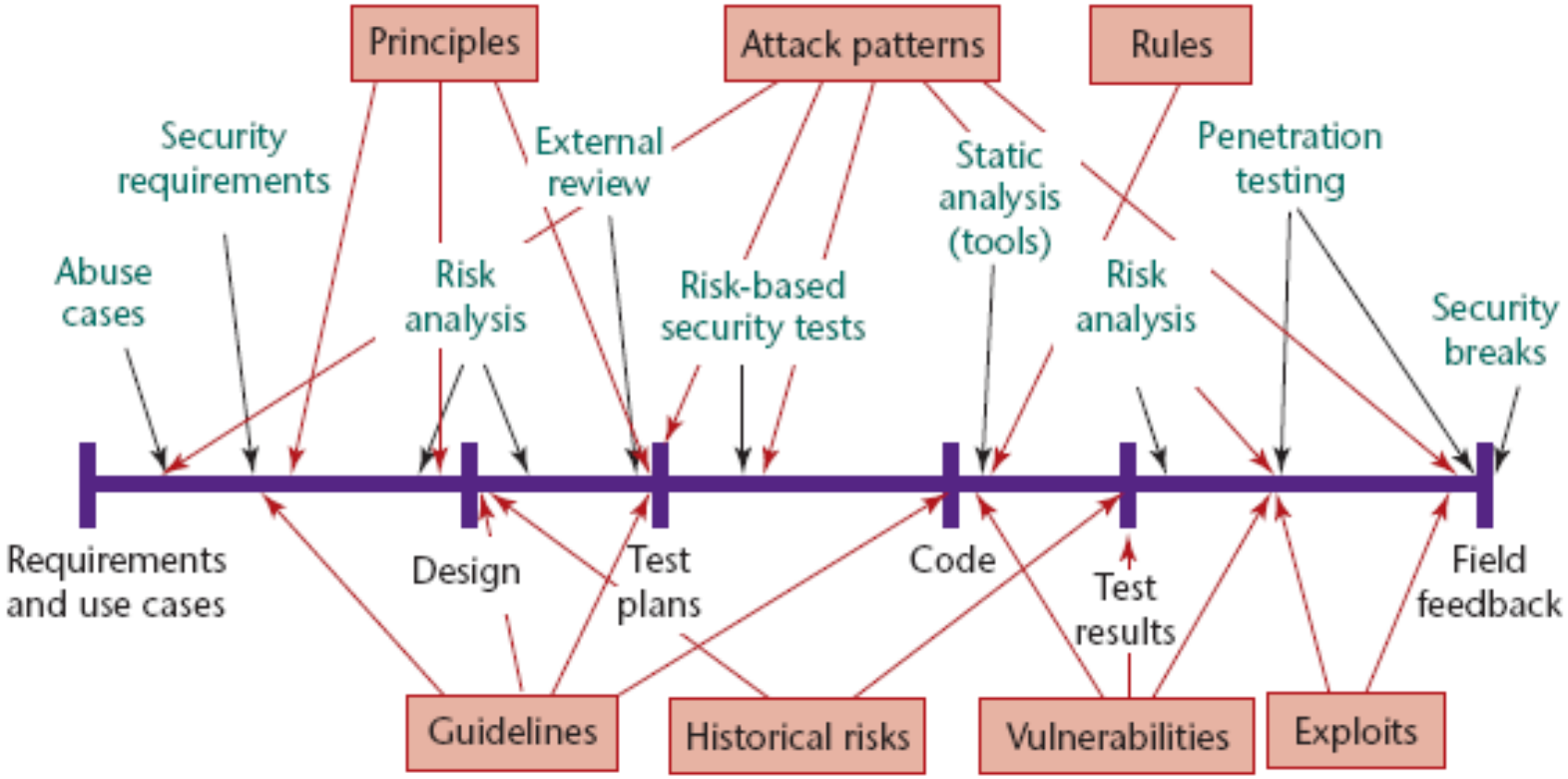
- *attack surface area reduced*
- *unused features turned off by default*
- *minimum privileges used*

Secure in Deployment

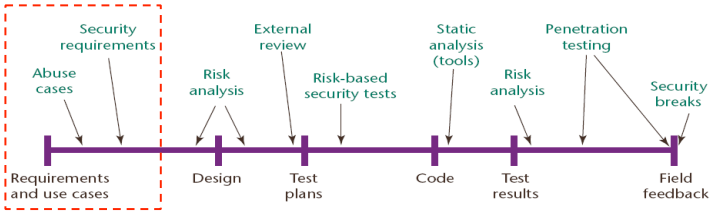
- *Protection: detect, defend, recover, manage*
- *Process: how to guides, architecture guides*
- *People: training*

Sumber : MSDN (Microsoft Developer Network)

Security Throughout Project Life Cycle



Security Requirement

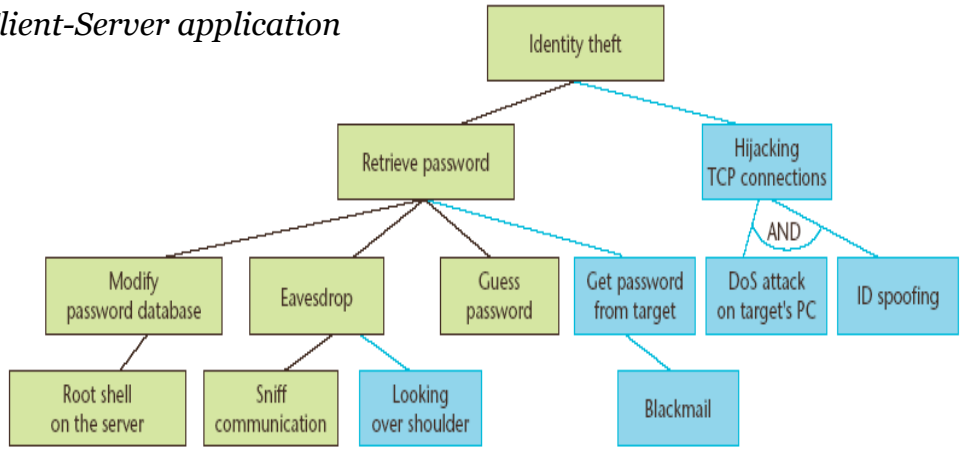


Mendefinisikan lingkungan *security* dan obyektif → **Mengembangkan model ancaman/serangan** → **Mensosialisasikan kebijakan *security***

Example:

A Client-Server application

Example:



Example:

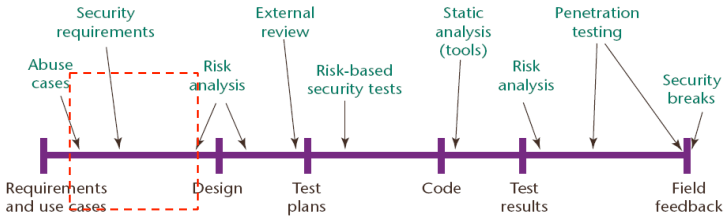
- *To use the presence or instant messaging services, users must be authenticated.*
- *Messages mustn't be tampered with.*
- *The message sender should be properly authenticated.*
- *Presence information shouldn't be tampered with.*
- *The subscriber database's privacy should be guaranteed.*
- *Message sending should be time-stamped security – if user A sends a message at time t to B, he can't modify the message so that B thinks it was sent at time t'.*

Security Requirement

- How to describe
  - Using natural language
  - Using diagrams
- Assignment
  - Create a security requirement for
    - internet banking web-based application
    - Sistem informasi kepegawaian

Contoh Security Requirements

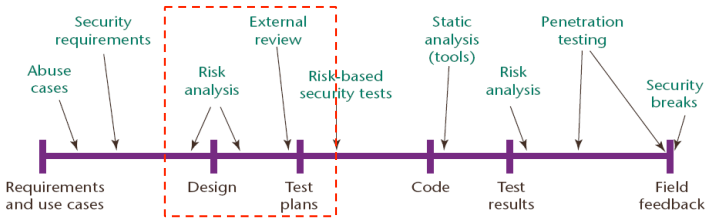
- Users must authenticate? Is anonymous access allowed? Is concurent access allowed?
- Password
  - Length? Strength?
  - Aging? (History?)
  - Failed attempts locked out? How many times? How to unlock?
  - Must be shown with asterisks (*)
- Is concurrent access allowed?
- Changes must be logged
- Audit log must provide enough data for forensic
- Data in transit, in process, in storage must be encrypted

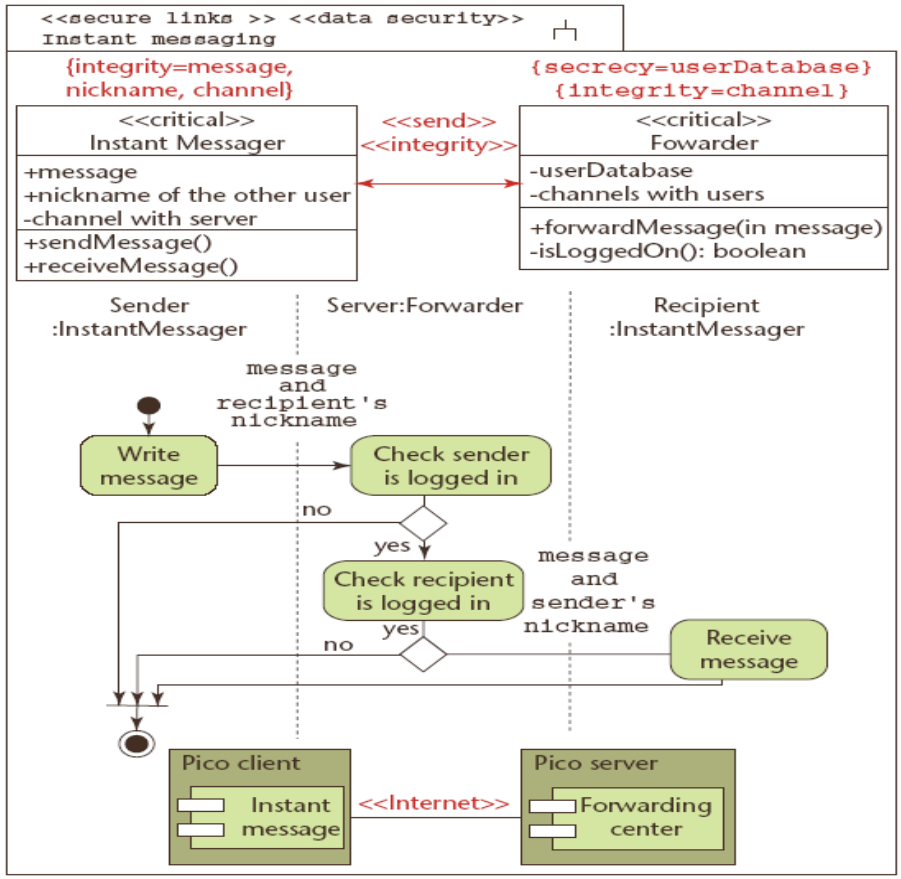


Secure Software Design

- Architectural Issues
  - Jika disain arsitektur software sudah memiliki lubang keamanan, maka implementasi tidak mengubah hal tersebut.
  - Analogi: bangunan yang didesain tanpa dinding di bagian belakang

Design Review





Contoh review terhadap desain: Software development by Example, IEEE Security & Privacy, July/August 2005

Contoh Security Design

- Requirement
  - Pengguna tidak boleh login dari dua (2) tempat yang berbeda (parallel/concurrent login)
    - Ada skenario pengujian login dari 2 IP yang berbeda pada saat yang bersamaan
    - Bagaimana caranya? Manual? Otomatis?
- Desain kendali
  - Cookies + nomor IP + jenis browser + identitas komputer lainnya
  - Pemberitahuan (dan pencatatan) kejadian

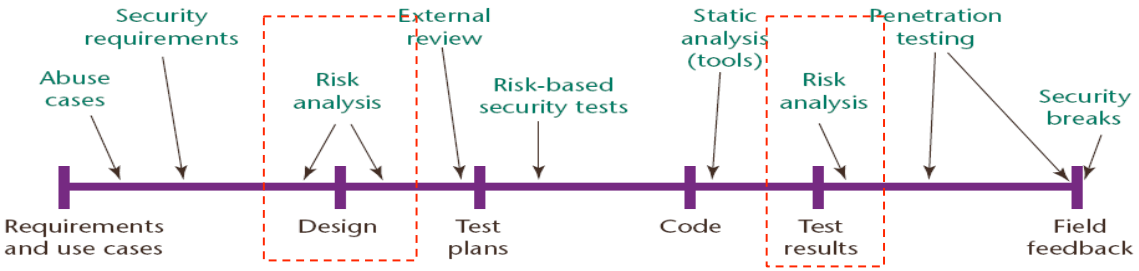
Implementasi: Secure Coding

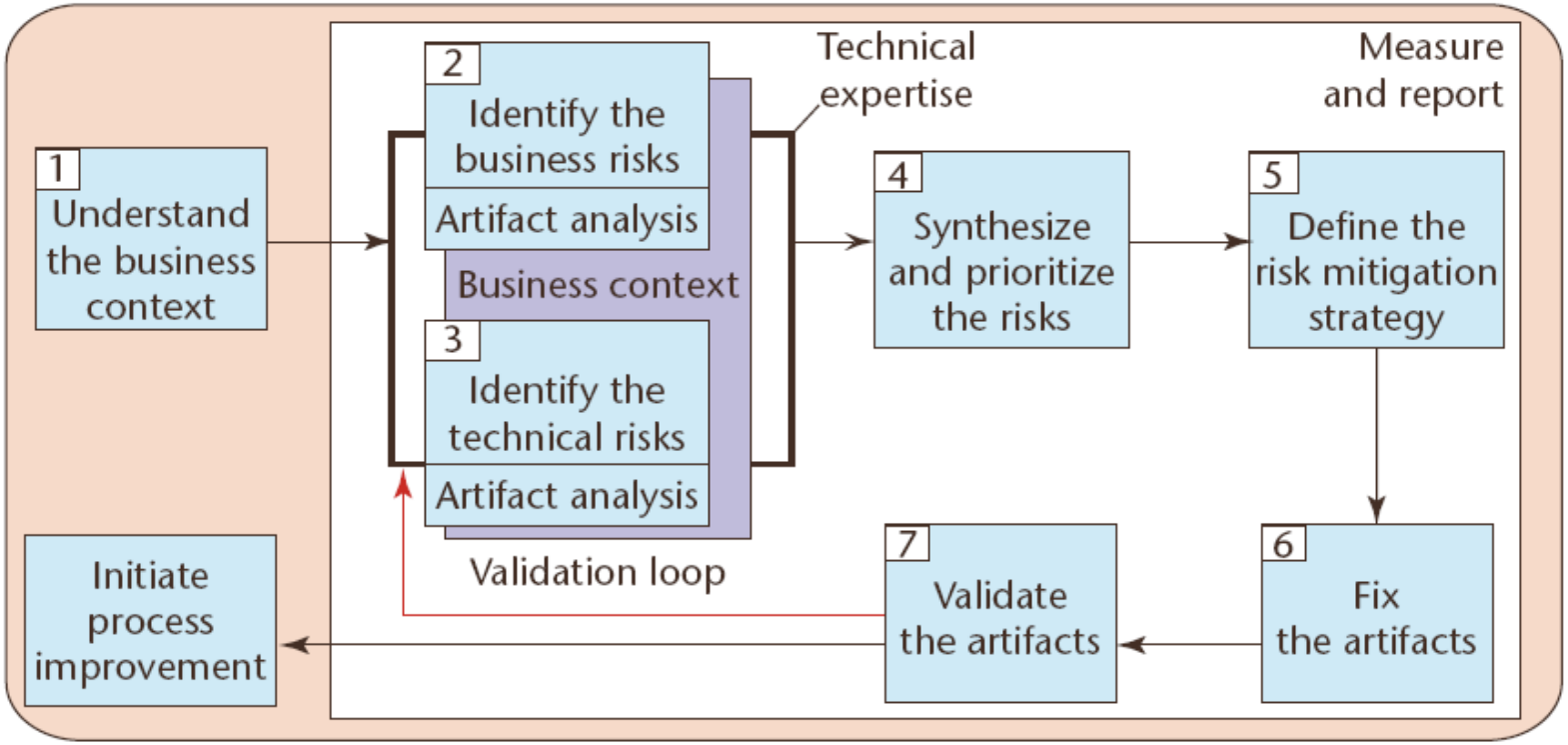
- Menentukan hasil dari implementasi
  - Analogi : Dinding yang dibangun dengan batu bata (beton) akan berbeda dengan tripleks (bedeng)
  - Ada beberapa tools yang dapat digunakan untuk menguji (static analysis tools)

Secure Code

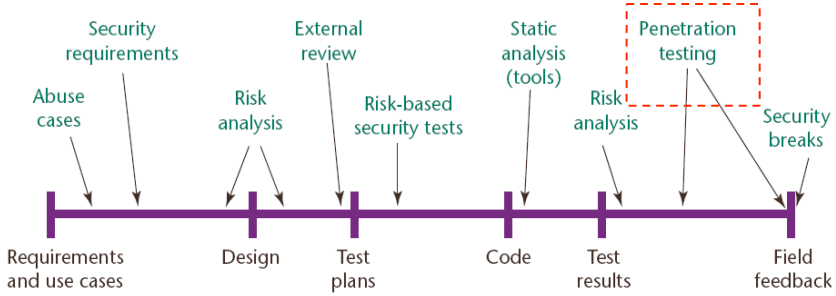
- Static analysis
  - Melakuan analisis terhadap source code
    - Bergantung kepada bahasa yang digunakan
    - Masih dibutuhkan banyak penelitian
- Dynamic analysis
  - Menjalankan software dan kemudian melakukan analisis keamanan
    - Melihat isi memory

Risk Analysis





Penetration Testing



Building a test plan

- *High level overview of the test cases*
- *How explanatory testing will be conducted*
- *Which component will be tested*

Executing test cases

- *Dependency testing*
- *User interface testing*
- *Design testing*
- *Implementation testing*

Reporting

- *Reproduction steps*
- *Severity*
- *Exploit scenarios*

Source: Application Penetration Testing, IEEE Security & Privacy, Jan/Feb 2005

"Improving the Security of Your Site by Breaking Into it”

(Dan Farmer/Wietse Venema, 1993)

http://www.fish.com/security/admin-guide-to-cracking.html

Blackbox vs. Whitebox

Blackbox

- Tidak ada pengetahuan awal mengenai sistem yang akan diuji.
- Penguji menentukan dulu lokasi dan coverage target.
- Menguji berbasis input dan output saja
  - Tanpa source code
  - Memberikan input yang tidak lazim

Whitebox

- Diberi pengetahuan lengkap mengenai sistem yang akan diuji
- Mencari lubang keamanan dengan menelusuri program :
  - Dengan source code
  - Identifikasi programming error
  - Mencari kelemahan (algoritma dan teknik implementasi)

Top Software Security Flaws

Buffer Overflow

Input validation bugs

Race condition

Penutup

- Keamanan perangkat lunak (software security) masih merupakan bidang yang baru
- Akan lebih banyak masalah terkait dengan software security
- Kemampuan mengembangkan software yang aman sangat dibutuhkan