計算機システム概論
計算機システムとセキュリティ
2011/5/27

門林雄基

NAIST 奈良先端科学技術大学院大学

講義のポイント

2

- セキュリティの基本概念とは？
- セキュリティを実現する技術とは？
- アクセス制御とは？ セキュリティ管理とは？
- セキュリティを考慮したソフトウェア開発とは？

 11/05/27

セキュリティの基本概念

3

- セキュリティは複合概念である
- 正確に問題を取り扱うためには具体的な用語を使ったほうがよい

- CIA Triad
  - Confidentiality, Integrity, Availability
- AAR
  - Authenticity, Accountability, Reliability

- どのセキュリティ属性を実現したいのか。
- そのためにどのような技術があるか。

セキュリティの基本概念（1）

4

- (1) 情報システムにおける情報保護
  - （犯罪行為、不正行為への対策）

可用性 (Availability)	• 破壊の防止
機密性・秘匿性 (Confidentiality)	• 盗聴の防止
一貫性・完全性 (Integrity)	• 改竄の防止

Source: OECD Guidelines for the Security of Information Systems, 1992

セキュリティの基本概念（2）

5

- (2) 情報システムにおける信用確保
  - （電子商取引、電子政府への対応）

追跡性 (Accountability)	• 事後否認の防止
信憑性・真正性 (Authenticity)	• なりすましの防止
信頼性 (Reliability)	• 誤動作の防止

Source: ISO/IEC IS 13335-1:2004, Information technology – IT Security techniques – Management of information and communications technology security -- Part 1: Concepts and models for information and communications technology security management

2x4 matrix で捉えるセキュリティ技術

まずイメージをつかもう

2x4 matrix で捉えるセキュリティ技術

7

	Hardware	Software
Human-Computer Interface		
Computer-Network Security		
Computer Security		
Information Security		

- セキュリティは総合科学
- 計算機システム概論の文脈で、それぞれの技術領域を見てみよう
  - 実装例によるイメージ

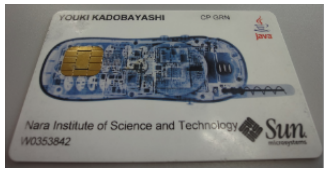
Security technologies in HCI hardware

8

- FeliCa, smart card
- バイオメトリック認証

NEC指紋認証ユニット PU900-10

Sony FeliCa



Smart card

Security technologies in HCI software

9

- パスワード生成プログラム

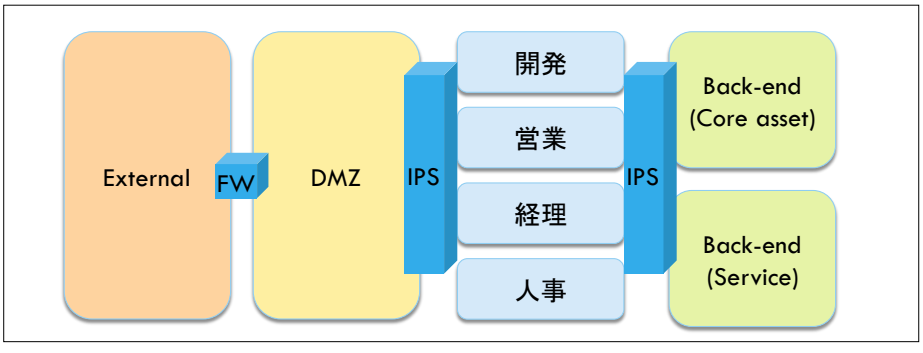
```
# pwgen 20  6
eWai6aiheengahWeidip aeWisoilah4udie8chud ub9oov8Eep9Aipu4oKee
xee9niochieMie8EKeiG ethe4aepohP2loo6Kaof nohtel4gae3aToopiohi
```

- ブラウザにおけるアドレスバー色表示

Hardware for Computer Network security

10

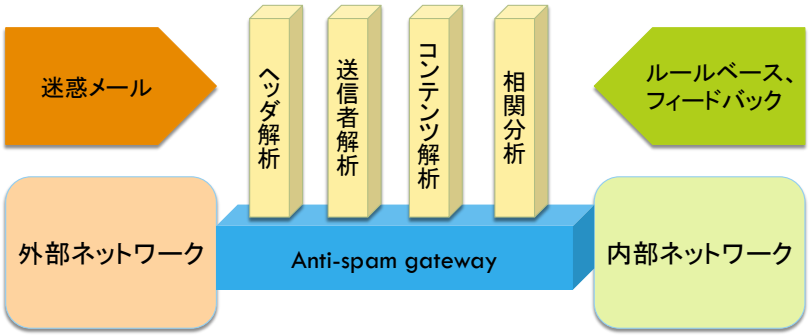
- Firewall
- Intrusion-protection system



Software for Computer Network security

11

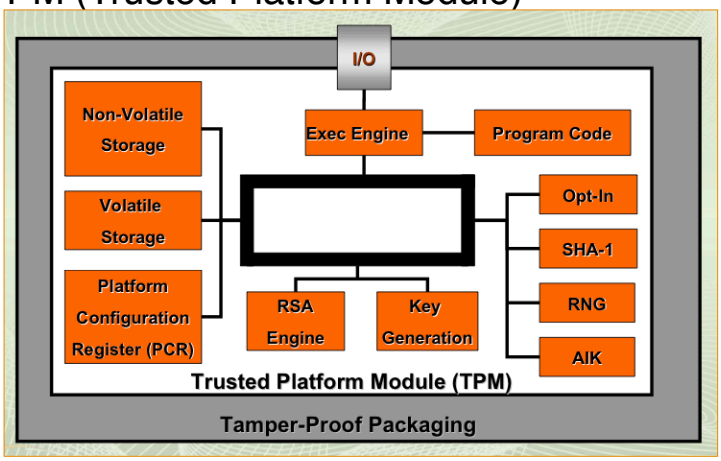
- Anti-spam software
- Intrusion-detection software



Hardware for computer security

12

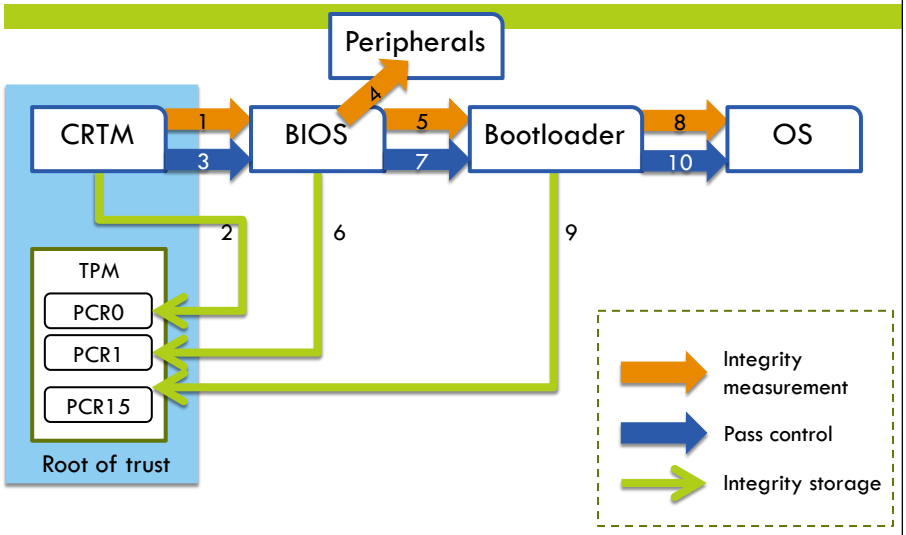
- TPM (Trusted Platform Module)



Source: Trusted Computing Group

TPMによるハードウェア信頼性の確保

13

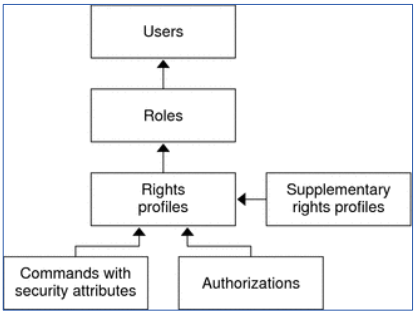


 11/05/27

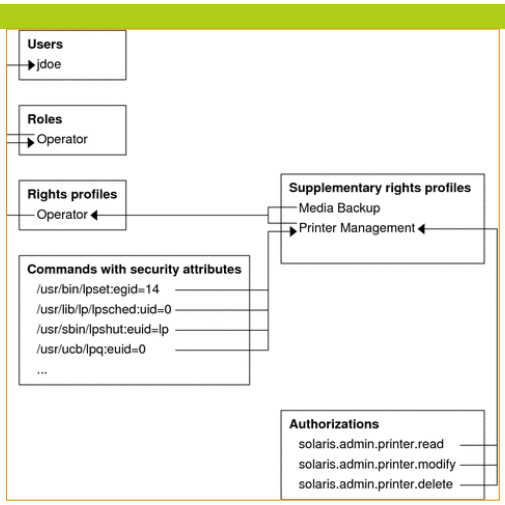
Software for computer security

14

- Secure OS
- Secure software



Solaris RBAC Element Relationships



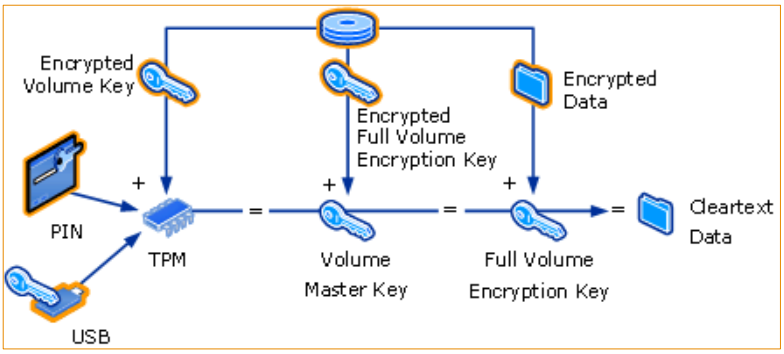
Source: OpenSolaris Role-Based Access Control

 11/05/27

Hardware for information security

15

- Encrypting storage



Source: BitLocker Drive Encryption Technical Overview, Microsoft TechNet, 2009

 11/05/27

Software for information security

16

- Encryption
- Digital signature

```
<Reference URI="#MyFirstManifest"
   Type="http://www.w3.org/2000/09/xmldsig#Manifest">
   <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
   <DigestValue>345x3rvEPO0vKtMup4NbeVu8nk=</DigestValue>
 </Reference>
...
<Object>
 <Manifest Id="MyFirstManifest">
   <Reference>  ...  </Reference>
   <Reference>  ...  </Reference>
 </Manifest>
</Object>
```

Source: RFC3275, XML-Signature Syntax and Processing

 11/05/27

セキュリティの機構的実現

17

認証 (Authentication)	• 利用者本人であることを担保する
権限管理 (Authorization)	• 利用者が権限を有することを担保する
保証 (Assurance)	• ハード・ソフトに欠陥がないことを担保する
軽減 (Mitigation)	• 問題事象を軽減する
電子署名 (Digital signature)	• 情報の一貫性を担保する
暗号化 (Encryption)	• 情報の秘匿性を担保する

これらの機構をいかに組み合わせて望みのセキュリティ属性を実現するか。
→ セキュリティポリシ

セキュリティポリシ

何を保護したい？

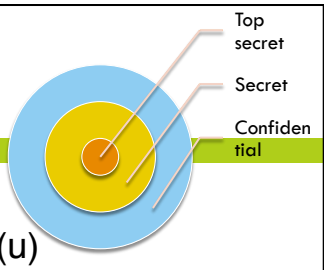
アクセス制御のモデル

19

- Bell-LaPadula model
  - "no read up, no write down.”
- Biba Model
  - "no write up, no read down."

Bell-LaPadula model

20

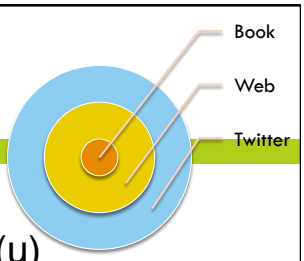


- 情報o：機密性レベル C(o)
- 利用者u：機密性取り扱いレベル C(u)
- 目的：異なる機密性レベルへの情報漏洩防止

- 高い機密性取り扱いレベル＝高い機密保持義務
- “No read up”
  - C(o) ≦ C(u) の場合のみアクセス許可
- “No write down”
  - C(o) ≦ C(p) かつ情報oにアクセスできる場合のみ情報pに書き込み可

Biba model

21



- 情報o：完全性レベル I(o)
- 利用者u：完全性取り扱いレベル I(u)
- 目的：低い完全性レベルからの情報汚染防止

- 高い完全性取り扱いレベル＝高い情報の信頼性
- “No write up”
  - I(o) ≦ I(u) の場合のみ書き込み許可
- “No read down”
  - I(u) ≦ I(o) の場合のみ読み出し許可

アクセス制御モデルの限界

頭でっかちの罠

22

- BLP, Biba は情報の取り扱いモデルとしては参考になるが、システムを安全に保つための方法論ではない。

- パスワードが推測可能であったとしたら？
- ソフトウェアに権限昇格につながる欠陥があったとしたら？
- システムが止まったら？

- 「想定外」が許されない時代のセキュリティポリシとは？

セキュリティ管理に関する産業標準

23

- IT統制、ITサービスマネジメント
  - COBIT
    (Control Objectives for Information and related Technology)
  - ITIL (IT Infrastructure Library)
- 情報セキュリティマネジメント
  - ISO/IEC 27002
    (Code of practice for information security management)
- 運用管理に対する網羅的な視点を提供する
  - チェックリスト
- 認証だけ取得して元の木阿弥、では意味がない
  - 日本では本末転倒か

チェックリストとしての標準

抜け・漏れのないセキュリティ管理を

24

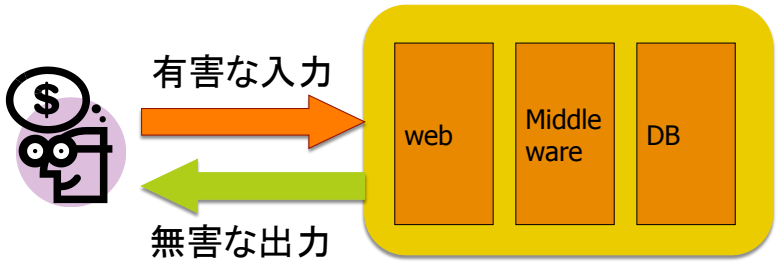
CobiT 4.1 Domain: Deliver and Support (DS) *(cont.)*			
DS5 Ensure Systems Security *(cont.)*			
CobiT 4.1 Control Objective	**Key Areas**	**ITIL V3 Supporting Information**	**ISO/IEC 27002:2005 Supporting Information**
DS5.2 IT security plan	• Translation of business, risk and compliance requirements into a security plan	• *SD 4.6.4 Policies/principles/basic concepts* • *SD 4.6.5.1 Security controls (high-level coverage, not in detail)*	• **5.1.1 Information security policy document** • **5.1.2 Review of the information security policy** • **6.1.2 Information security co-ordination** • **6.1.5 Confidentiality agreements** • **8.2.2 Information security awareness, education and training** • **11.1.1 Access control policy** • **11.7.1 Mobile computing and communications** • **11.7.2 Teleworking**
DS5.3 Identity management	• Identification of all users (internal, external and temporary) and their activity	• *SO 4.5 Access management*	• **11.2.3 User password management** • **11.3.1 Password use** • **11.4.1 Policy on use of network services** • **11.5.1 Secure logon procedures** • **11.5.2 User identification and authentication**

Source: Aligning CobiT®4.1, ITIL®V3 and ISO/IEC 27002 for Business Benefit, ITGI/OGC

排斥型アプローチの限界

開放型システムにおけるアクセス制御というナンセンス

25



Web: 一般大衆を相手にしたシステム

正規ユーザでも情報漏洩を引き起こすことを目的としている可能性

 11/05/27

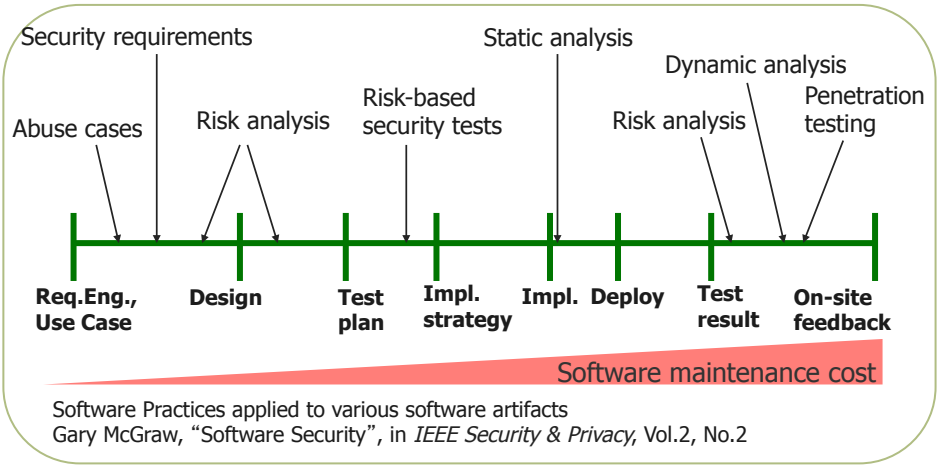
ソフトウェア開発における セキュリティ

それでも守るためには

開発フェーズと対策コスト

27

- 設計段階からセキュリティを考慮することで
  最終フェーズにおける対策コストを低減可能

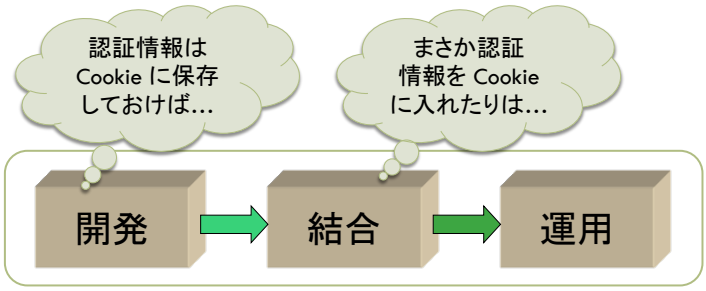


Software Practices applied to various software artifacts
Gary McGraw, “Software Security”, in *IEEE Security & Privacy*, Vol.2, No.2

工程間のリスク

28

- 開発、システム結合、運用のコミュニケーションリスク
  - 各工程で暗黙の前提
  - じゅうぶんなコミュニケーションがとられない場合が多く、
    前提の違いが脆弱性につながるケースも

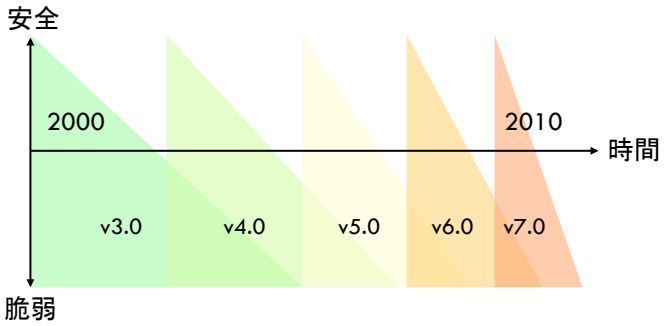


 11/05/27

ソフトウェアの脆弱性と時間軸

29

- 加速度的に増えるソフトウェアの脆弱性
- ファジング → 脆弱性の発現速度が加速
- プログラム修正で対処可能なスピードを超える危険性



 11/05/27

ソフトウェア開発における セキュリティ成熟度モデル

30

- 順にみていこう
- BSIMM2 (Building Security In Maturity Model)
  - Gary MacGraw, Brian Chess, Sammy Migues
- SAMM (Software Assurance Maturity Model)
  - Pravir Chandra & OWASP

 11/05/27

BSIMM2

Building Security In Maturity Model

31

- セキュアソフトウェア開発ライフサイクルを参照
- 開発プロセスに対する俯瞰的視点

domain	practice	business goals
Governance	Strategy and Metrics	Transparency of expectations, Accountability for results
	Compliance and Policy	Prescriptive guidance for all stakeholders, Auditability
	Training	Knowledgeable workforce, Error correction
Intelligence	Attack Models	Customized knowledge
	Security Features and Designs	Reusable designs, Prescriptive guidance for all stakeholders
	Standards and Requirements	Prescriptive guidance for all stakeholders
SSDL Touchpoints	Architecture Analysis	Quality control
	Code Review	Quality control
	Security Testing	Quality control
Deployment	Penetration Testing	Quality control
	Software Environment	Change management
	Vulnerability Mgmt and Change Management	Change management

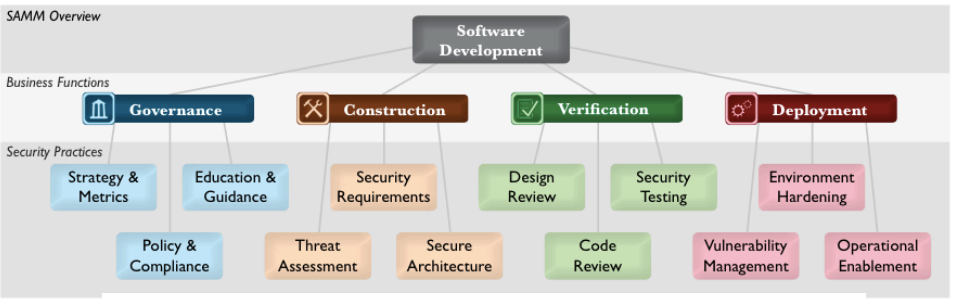
Source: BSIMM2 (Building Security In Maturity Model), May 2010.

SAMM

Software Assurance Maturity Model

32

- 開発現場で実施可能な検討項目、プロセス改善などを成熟度別に整理



Source: SAMM 1.0 (Software Assurance Maturity Model) www.opensamm.org

まとめ

33

- セキュリティは複合概念
  - 基本6概念への分解
- 2x4 matrix で捉えるセキュリティ技術
- アクセス制御のモデル
- セキュリティ管理
  - 運用における網羅的な視点
- セキュリティを考慮したソフトウェア開発
  - コスト認識、リスク認識、方法論