보안공학이란?
(Security Engineering)

고려대학교 (Korea Univ.)

사이버국방학과 · 정보보호대학원 (CIST)

보안성분석평가연구실 (Security Analysis aNd Evaluation Lab.)

김 승 주 (Seungjoo Kim)

www.kimlab.net

Who am I?

보안성분석평가연구실

Security Analysis aNd Evaluation Lab

www.KimLab.net / www.SecEng.net

김승주 교수 (skim71@korea.ac.kr)

로봇융합관 306호

주요 경력 :

1990.3~1999.2) 성균관대학교 공학 학사-석사-박사
1998.12~2004.2) KISA 암호기술팀장 및 CC평가1팀장
2004.3~2011.2) 성균관대학교 정보통신공학부 부교수
2011.3~현재) 고려대학교 사이버국방학과·정보보호대학원 정교수
Founder / Advisory Director of SECUINSIDE

前) 육군사관학교 초빙교수
前) 선관위 DDoS 특별검사팀 자문위원
前) SBS 드라마'유령'및 영화'베를린'자문

現) 한국정보보호학회 이사
現) 대검찰청 디지털수사 자문위원
現) 개인정보분쟁조정위원회 위원

- '96: Convertible group signatures (AsiaCrypt)
- '97: Proxy signatures, revisited (ICICS): 630회 이상 인용
- '06: 국가정보원 암호학술논문공모전 우수상
- '07: 국가정보원장 국가사이버안전업무 유공자 표창
- '12,'16: 고려대학교 석탑강의상
- '13: Smart TV Security (Black Hat USA): 스마트TV 해킹(도청·도촬) 및 해적방송 송출 시연

연구분야

- Security Eng. for High-Assurance Trustworthy Systems (e.g. End-to-End Provably Trustworthy Kernel)
- Recent Security Threat Analysis and Security Evaluation (e.g. CMVP, CC, ISMS, C&A)
- All Areas of Security, from Crypto to Hacking, and Policy

주요 연구성과

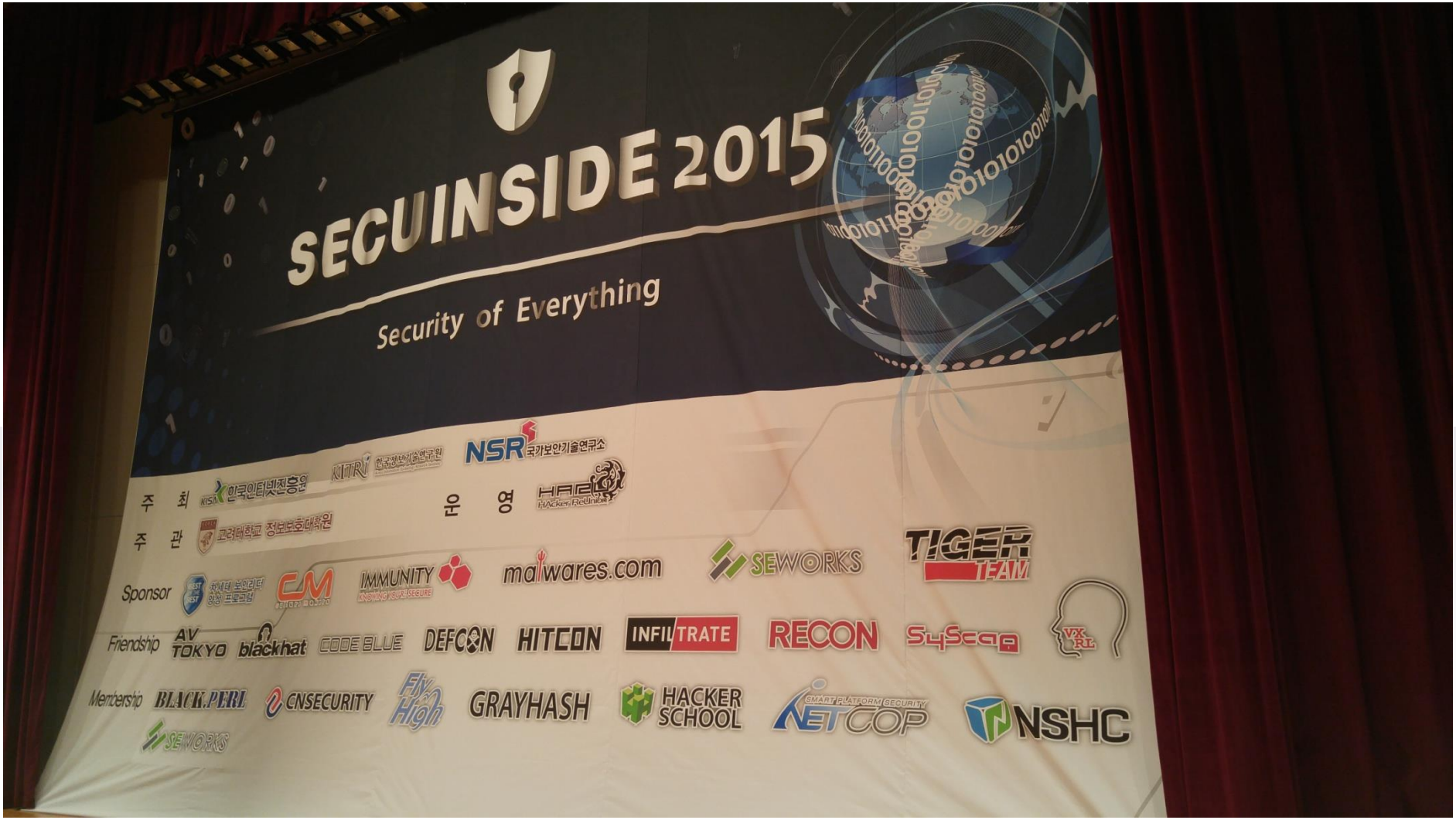
MBC 뉴스데스크
(2013.5.10.)

CyKoR @ DEFCON CTF 2015

(**Advisor :** Seungjoo Kim)

KOREA UNIVERSITY

(사)HARU & SECUINSIDE



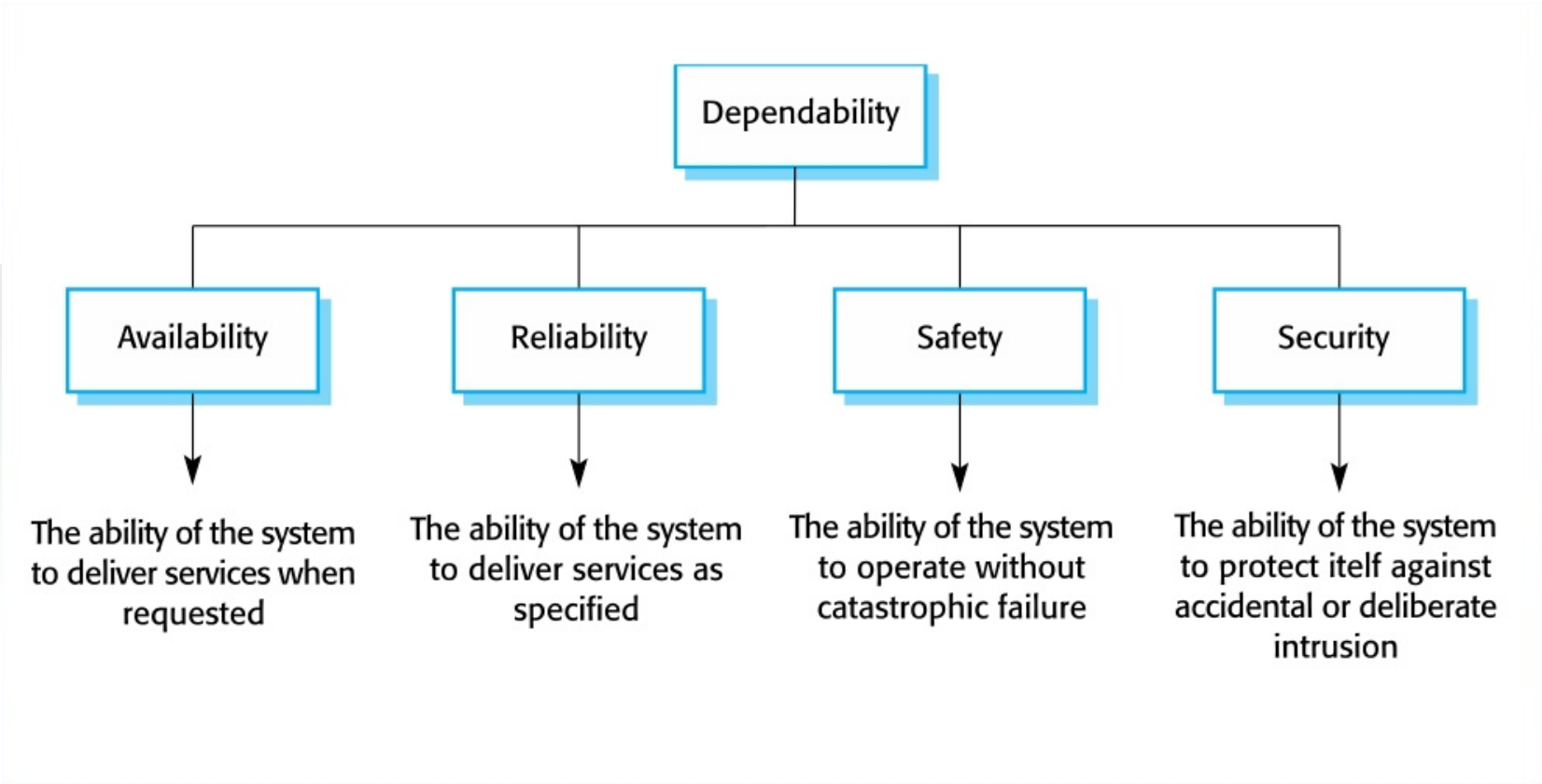
(**Founder & Board Member :** Seungjoo Kim, 2011)

KOREA UNIVERSITY

Introduction to Security Eng.

Security Engineering (보안공학)

- Security engineering is about building **dependable** (or **trustworthy**) systems.



☞ 'Dependability' is interchangeably used for 'trustworthiness'
(Sommerville, I.: "Software Engineering", Addison-Wesley, 6. edn., 2001, ISBN 0-201-39815-X)

Security Engineering (보안공학)

- **Trustworthiness** (or **Dependability**) is assurance that a system deserves to be trusted - that it will perform as expected despite **environmental disruptions**, **human and operator error**, **hostile attacks**, and **design and implementation errors**. Trustworthy systems reinforce the belief that they will continue to produce expected behaviour and will not be susceptible to subversion.

(Source : The "Trust in Cyberspace" report of the United States National Research Council)

Security Engineering (보안공학)

- **Financial System**
  - High security + Medium reliability + No safety
- **DB of Medical Records**
  - Medium security + Medium reliability + Medium safety
- **Air Traffic Control System**
  - Medium security + High reliability + High safety
- **Automobile**
  - Low (but now medium!) security + High reliability + High safety

5 Steps for Developing Dependable S/W

1. Define "**goals**" or "**properties**" (i.e., what you want the program to satisfy)
2. Design **algorithms/protocols**
3. Make **standards**
4. Generate **source code**
5. Compile to **machine code** (i.e., what actually runs)

Problems for STEP 1.

1. Define "**goals**" or "**properties**" (i.e., what you want the program to satisfy)

How to identify and define goals correctly?

✓ By using "**Threat Modeling**" & "**Security Policy Modeling (SPM)**"

Problems for STEP 2.

2. Design **algorithms/protocols**

How to check if your algorithm or protocol satisfy the goals of STEP 1?

✓ By "**hand-proof**" or "**machine-checked proof**"

Problems for STEP 3.

3. Make **standards**

There might be specification mismatch between STEP 2 and STEP 3.

✓ We need **equivalence proof** to address the "gap" between the abstract algorithm/protocol and more concrete standard specification

Problems for STEP 4 ~ STEP5.

4. Generate **source code** and compile it into **machine code**

Program might incorrectly implement the standard of STEP 3.

Also, we can't be sure about the compiler!

✓ By using machine-checked proof tools such as "**Verified Software Toolchain**".

Assurance Levels

1. Define "**goals**" or "**properties**" (i.e., what you want the program to satisfy)

In many cases, "provable security in cryptography" means only "design assurance"!
i.e., the proposed algorithm/protocol satisfies certain security requirements

2. Design **algorithms/protocols**
3. Make **standards**
4. Generate **source code**
5. Compile to **machine code** (i.e., what actually runs)

Assurance Levels

1. Define "**goals**" or "**properties**" (i.e., what you want the program to satisfy)
2. Design **algorithms/protocols**
3. Make **standards**

Common Criteria, even at EAL7, relies on testing (not mathematical proof!). There is no proof that security properties hold for the actual implementation (i.e., code proof).

4. Generate **source code**
5. Compile to **machine code** (i.e., what actually runs)

Assurance Levels

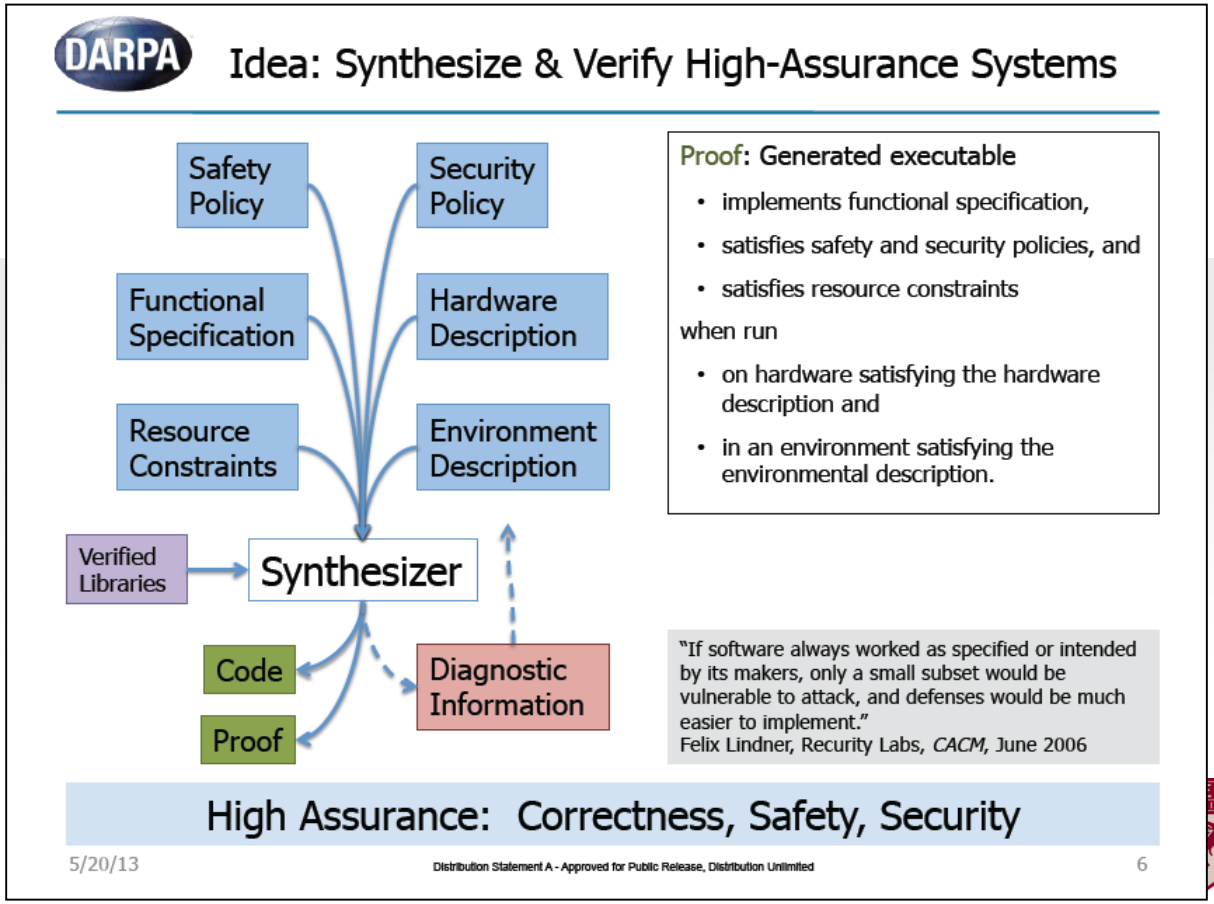
1. Define "**goals**" or "**properties**" (i.e., what you want the program to satisfy)
2. Design **algorithms/protocols**
3. Make **standards**
4. Generate **source code**

We call this as "High-Assurance (End-to-End Provably Dependable) Systems"
(e.g.) DARPA's HACMS(Hisg-Assurance Cyber Military Systems) Program, NICTA's seL4 Microkernel, etc

5. Compile to **machine code** (i.e., what actually runs)

DARPA's HACMS

- **Hack-Proof** Drones Possible with DARPA's HACMS(**High Assurance** Cyber Military Systems) Technology



KOREA UNIVERSITY

NICTA's seL4 Microkernel

TechWorld

Home | Technology | Reviews | Tools & Resources | Whitepapers | Careers | Login | Search Techworld

NICTA demos drone OS using the seL4 microkernel

Operating system will be released as open source on 29 July (AEST)

Rebecca Merrett (Techworld Australia) on 03 July, 2014 12:20

0 Comments

NICTA has created a video that demonstrates how its seL4 microkernel, the "world's most highly assured OS", can be used to operate drones.

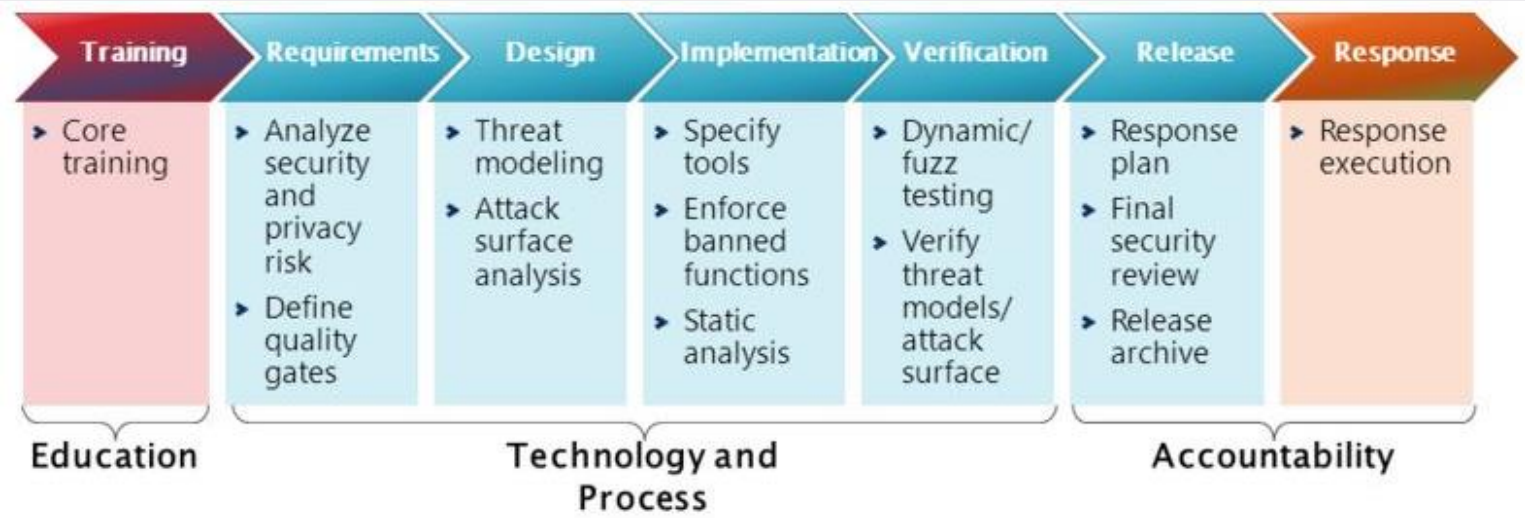
OREA NIVERSITY

Why Formal Proof(Verification)?

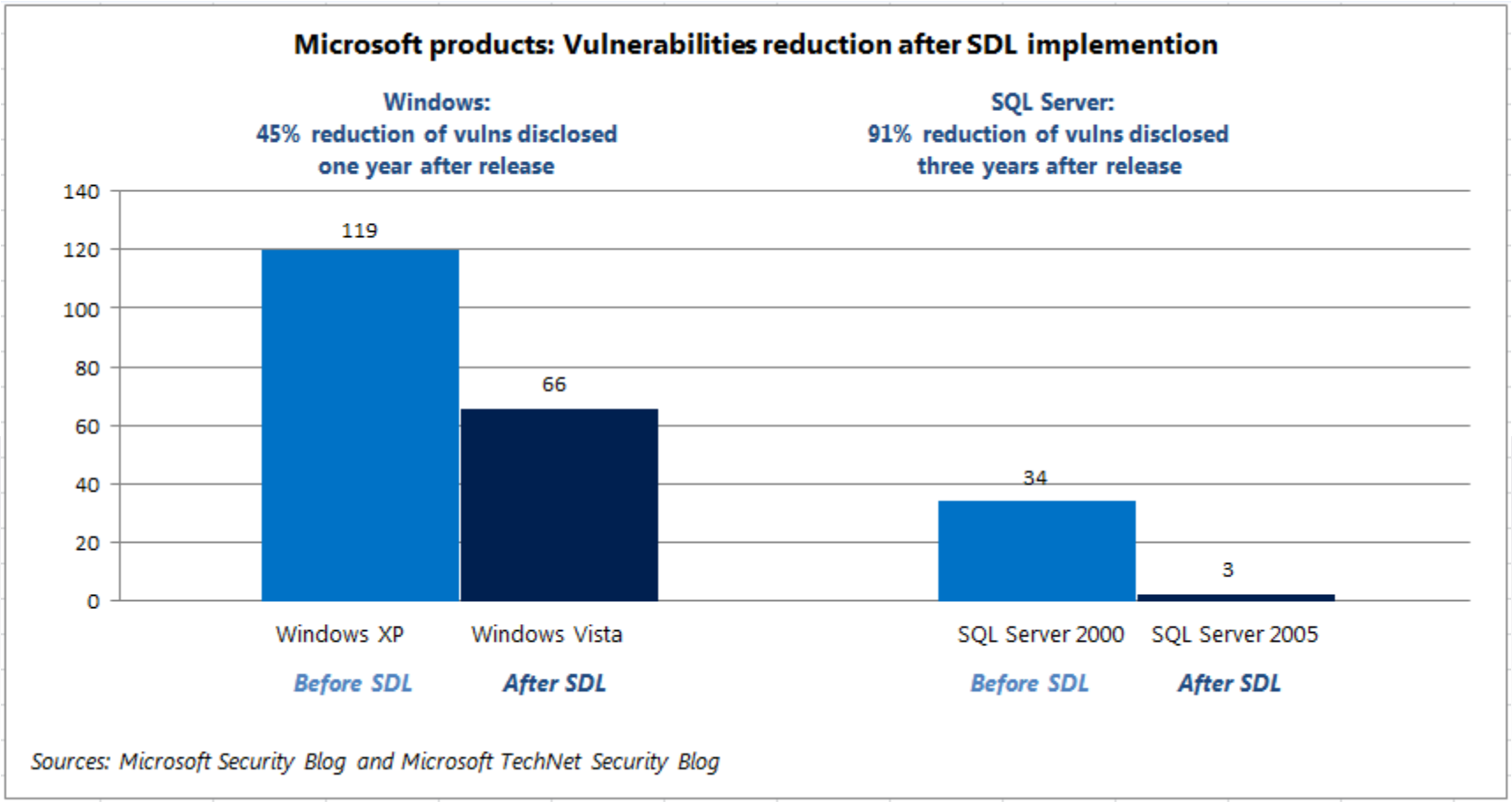
- **Simulation** and **test** cannot handle all possible cases (only **some** possible test vectors)
- **Simulation** and **test** can prove the **presence** of bugs, rather than their **absence**
- **Formal proof** conducts exhaustive exploration of **all** possible behaviors
  - If verified correct, **all** behaviors are verified
  - If verified incorrect, a **counter-example** (proof) is presented

If It Is Not Provable…

- Mathematical proofs are best! But if it is not achievable, we should follow the **well-defined software development processes**!!
  - (e.g.) Microsoft's SDL(Security Development Lifecycle)



If It Is Not Provable… - MS SDL



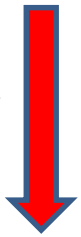
KOREA UNIVERSITY

If It Is Not Provable… - CC Evaluation

Common Criteria

If It Is Not Provable… - CC Evaluation

Scope, Depth, Rigor



Common Criteria	Requirements	Functional Specification	HLD	LLD	Implementation
EAL 1	Informal	Informal	Informal	Informal	Informal
EAL 2	Informal	Informal	Informal	Informal	Informal
EAL 3	Informal	Informal	Informal	Informal	Informal
EAL 4	Informal	Informal	Informal	Informal	Informal
EAL 5	Formal	Semiformal	Semiformal	Informal	Informal
EAL 6	Formal	Semiformal	Semiformal	Semiformal	Informal
EAL 7	Formal	Formal	Formal	Semiformal	Informal
End-to-End Proof	Formal	Formal	Formal	Formal	Formal

High-Assurance Cyber System

KOREA UNIVERSITY

Anything else?

- For example, some assurance methods are applicable only to processes (i.e., ISO/IEC 21827),
- Others are applicable to products (i.e., ISO/IEC 15408 Information technology - Security techniques - Evaluation criteria for IT security) and
- Others are applicable to security management (i.e., ISO/IEC 27001 Information technology - Security techniques - Information security management systems - Requirements).

모의해킹(**Pen Test**)이 본질적으로 부수는(**break**) 걸 연구하는 분야라면 보안공학(**security engineering**)은 만드는 걸 연구하는 학문입니다.

그냥 만드는게 아니라 본인이 만든 물건이 어떤 환경에서 얼마만큼 안전한지 가급적 정량적으로 입증할 수 있게끔 만드는걸 배우는 학문입니다. **CC**(**Common Criteria**)를 비롯한 많은 평가기법들도 본질적으로는 이러한 정신을 계승합니다.

그렇게 만들어 봤자 환경이 조금만 바뀌면 또 깨질수 있는것 아니겠냐구요?

맞습니다. 그래도 자기가 만든게 도대체 어느정도 안전한지 아무것도 모르는 것 보다는 낫잖아요! 창과 방패 운운하면서리.. ^^;;

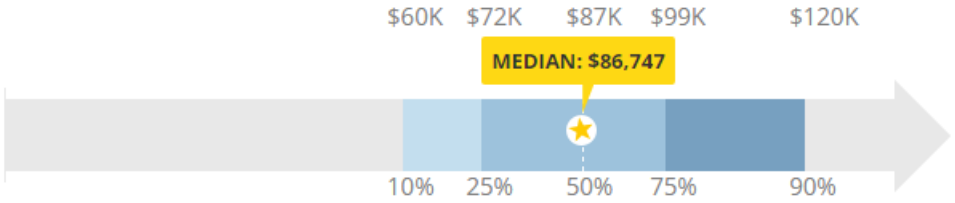
Your system is secure?

Prove it!

Information Assurance Engineer

Information Assurance Engineer Salary (United States)

Earnings for Information Assurance Engineers in the United States come in at around $87K annually on average. Final cash compensation to Information Assurance Engineers varies
Read More



Add this chart to your site: 640px 300px

+ city + experience + skill job

Show Hourly Rate

National Salary Data (?)	
Salary	$59,938 - $120,248
Bonus	$0.00 - $5,022
Profit Sharing	$0.00 - $10,067
Total Pay (?)	**$60,172 - $123,321**

$0 $50K $100K $150K

© Payscale, Inc. @ www.payscale.com

Country: United States | Currency: USD | Updated: 12 Jan 2016 | Individuals Reporting: 339

Skills That Affect Information Assurance Engineer Salaries

Security Policies and Procedures	▲7%
Security Risk Management	▲7%
IT Security & Infrastructure	▲6%
Security Testing and Auditing	▲2%
Computer Security	0%
National Average	**$87,000**

KOREA UNIVERSITY

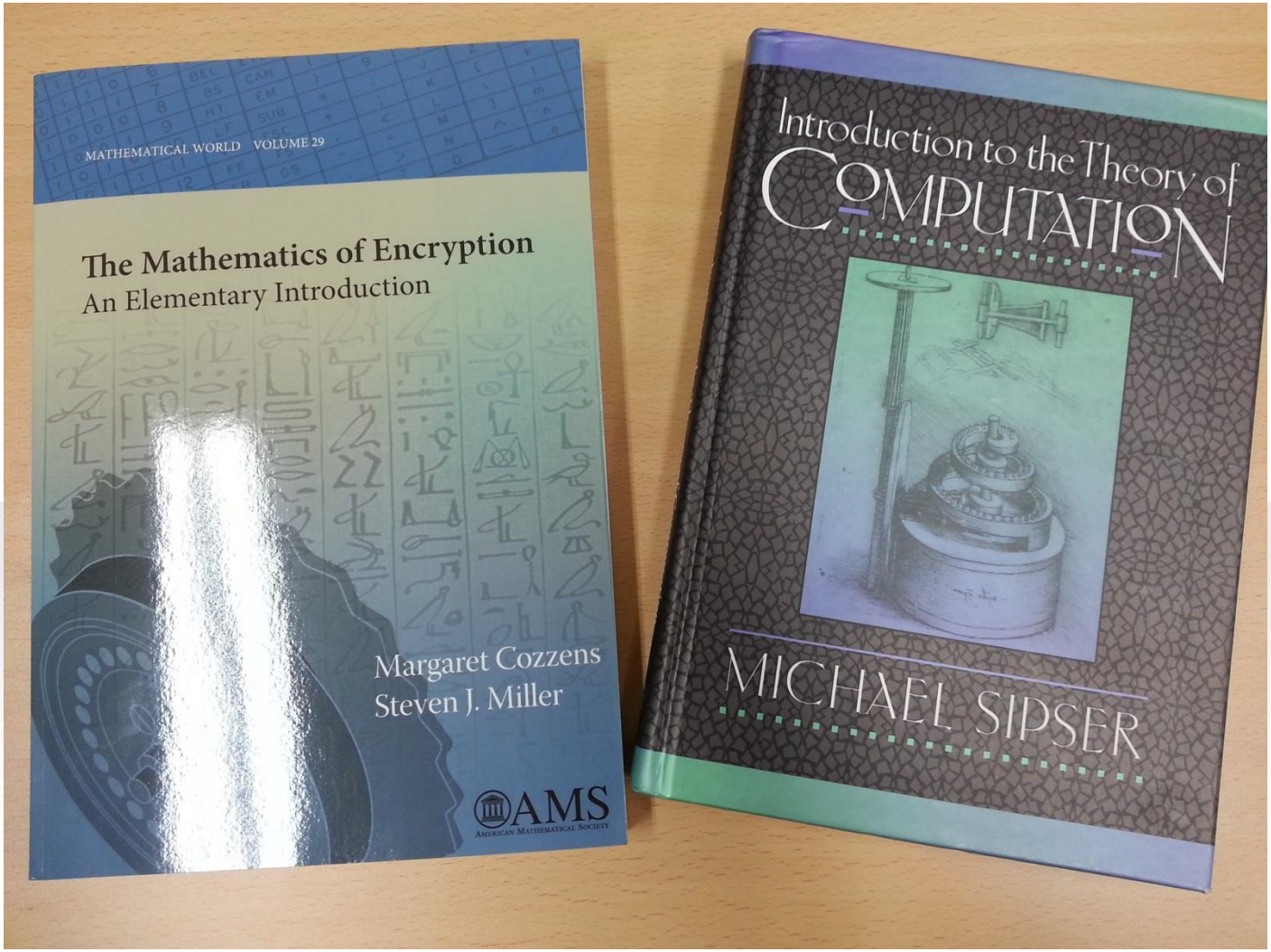
References

KOREA UNIVERSITY

References

KOREA UNIVERSITY

References



KOREA UNIVERSITY

References

KOREA UNIVERSITY

ICCC (International CC Conference)

ICMC (International Crypto Module Conf)

KOREA UNIVERSITY

Term Project

1) Select smart phone, car or any other IoT device in your home.
2) Analyze 'attack surface' of it by using Threat-Risk Modeling as a tool. (Firstly you should define Attacker Model)
3) Lead 'security requirements' from the above results.
4) Check out the related 'compliance and policy'.
5) Suggest 'security solutions' meeting (3) and (4).
6) Provide the 'rationale' that your solutions suggested in (5) are correct. (i.e., Show the design assurance & implementation assurance)

보안공학이란?
(Security Engineering)

고려대학교 (Korea Univ.)

사이버국방학과 · 정보보호대학원 (CIST)

보안성분석평가연구실 (Security Analysis aNd Evaluation Lab.)

김 승 주 (Seungjoo Kim)

(FB) www.fb.com/skim71 (Twitter) @skim71

고려대학교 정보보호대학원 KOREA UNIVERSITY