홍익대학교
HONGIK UNIVERSITY

Network Security

<CH 1>

Youn Kyu Lee
Hongik University

Brief Bio

- **Education**
  - B.S. and M.E. in CS from **Korea University**
  - Ph.D. in CS from University of **Southern California**
- **Work Experience**
  - AI&SW center at **Samsung Advanced Institute of Technology**
    - Fingerprint anti-spoofing for mobile devices
    - Fingerprint authentication system for mobile devices
  - Dept. of Information Security at **Seoul Women's University**
- **Research Interests**
  - Software Analysis / Architecture, Software Security
  - Artificial Intelligence, Deep Learning
- **Feel Free to Contact!**
  - **aiclasshongik@gmail.com**
  - younkyul@hongik.ac.kr

Information Security

This Lecture

- give you a thorough understanding of information security technology
  - what should be protected?
  - how can we protect?
  - from what?
  - how can we know when we're done?
- tell you how these techniques have been applied to a wide range of real-world security systems

This Lecture

- Cryptography
- Authentication and Authorization
- Protocols
- Software Security
- Mobile Security
- AI security, Bitcoin, Blockchain Technologies

Evaluation

- **Mid-term exam: 40%**
- **Final-term exam: 40%**
- **Assignments: 20%**

Schedule

5. 강의 내용 및 일정 Course Schedule			
Week	강의 및 실습내용 Topics, Assignments, Required Studies	교재내 범위 Readings	기타 Other Objectives
1	Introduction		
2	Cryptography		
3	Cryptography		
4	Cryptography		
5	Authentication		
6	Protocols		
7	Authorization		
8	Mid-term Exam		
9	Network Security		
10	Network Security		
11	Software Security		
12	Software Security		
13	AI Security		
14	Cryptocurrency and Bitcoin		
15	Course Summary / Final-term Exam		
참고사항 Note	수강생들의 이해 수준에 따라 다루는 내용 및 순서 등이 수정될 수 있음		

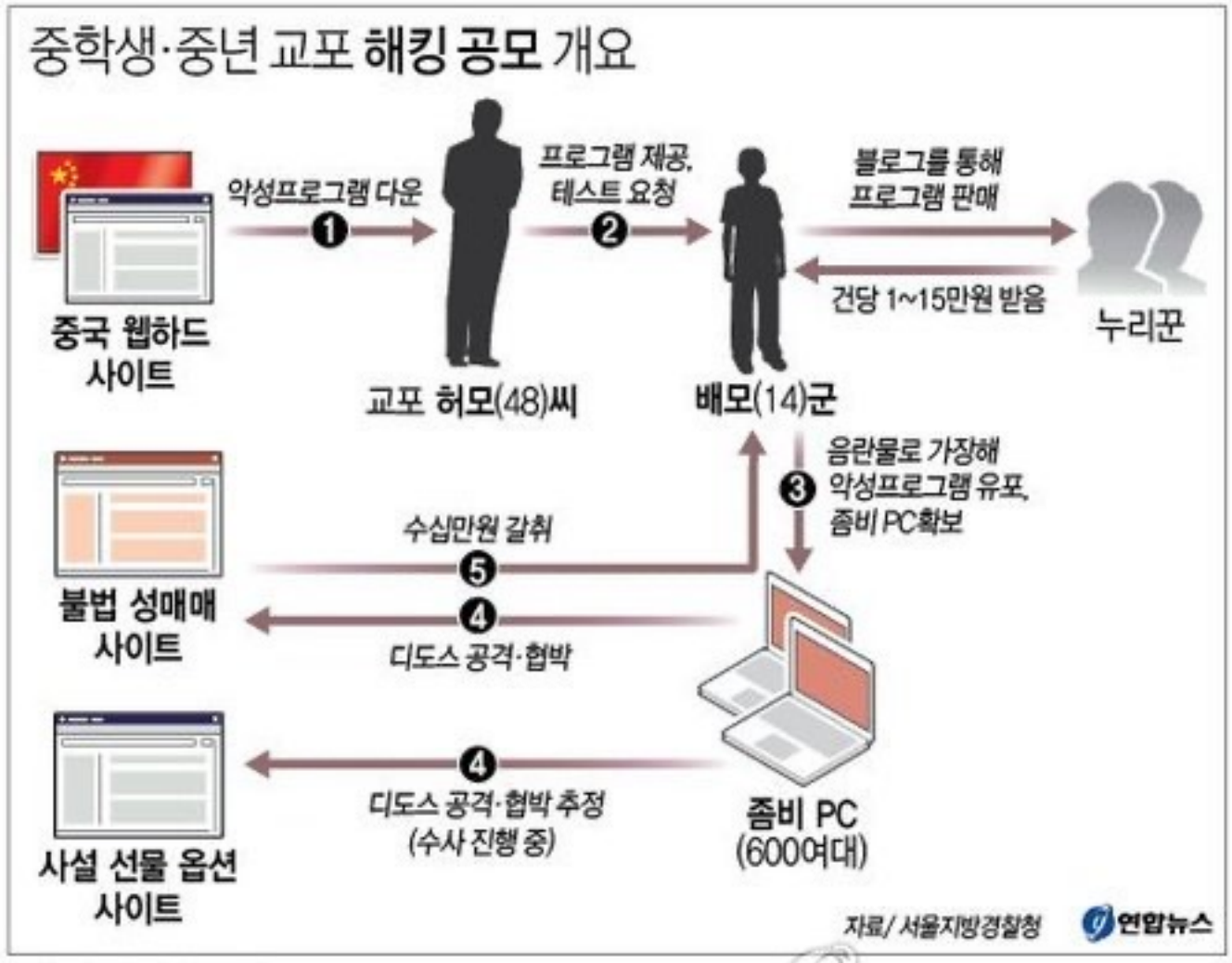
Q&A

aiclasshongik@gmail.com

Information Security

Information Security

Information Security



장예진 기자 / 2013704
@yonhap_graphics(트위터)

Information Security

발생 시기		거래소 명	피해 규모	피해 원인	결과	발생 국가
'18	9월	자이프	667억 원	해킹	매각	일본
	6월	빗썸	350억 원	이메일 악성코드 추정	피해 보상	한국
	6월	코인레일	450억 원	이메일 악성코드 추정	피해 보상	한국
	2월	비트그레일	1,921억 원	해킹	파산	이탈리아
	1월	코인체크	5,700억 원	해킹	매각	일본
'17	12월	유빗	172억 원	핫월렛 해킹	파산	한국
	12월	나이스해시	650억 원	핫월렛 해킹	피해 보상	슬로베니아
	9월	코인이즈	21억 원	핫월렛 해킹	피해 보상	한국
	7월	코인대시	79억 원	ICO 입금 주소 변경	피해 보상	미국
	6월	빗썸	30억 원	직원 자택 PC 해킹	피해 보상	한국
	4월	야피존	55억 원	개인키 도난	패쇄	한국
'16	8월	비트파이넥스	722억 원	개인키 도난	피해 보상	홍콩
	6월	더다오	510억 원	무한 환불 공격	하드포크	독일
'15	1월	비트스탬프	55억 원	악성코드	피해 보상	룩셈부르크
'14	10월	민트팔	25억 원	미상	파산	영국
	2월	마운트 곡스	4,700억 원	DDos	파산	일본

Information Security

Security Engineering

Security engineering is about building systems to remain dependable in the face of malice, error and mischance. As a discipline, it focuses on the tools, processes and methods needed to design, implement and test complete systems, and to adapt existing systems as their environment evolves.

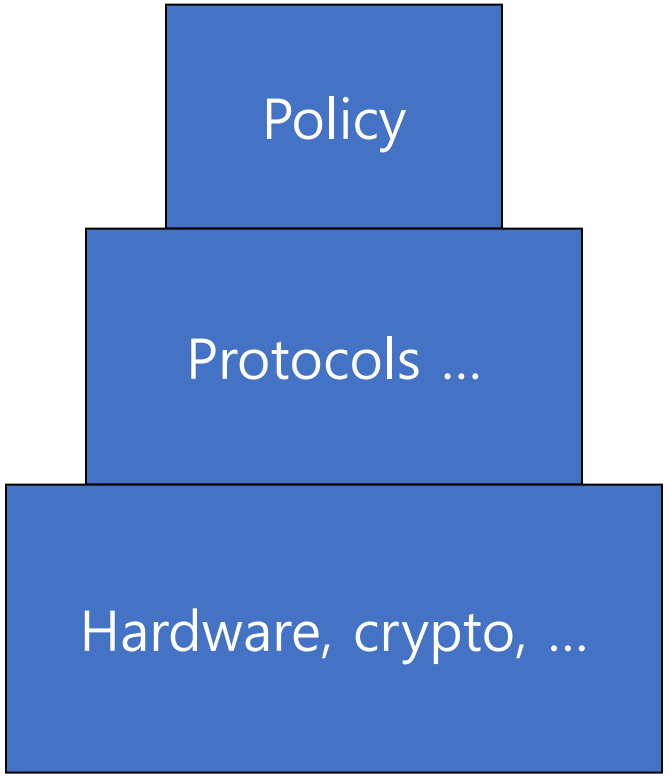
A salesman says:

"No need to worry, our product is 100% secure. All data is encrypted with a 256-bit key. It will take billions of years to crack our product."

- You should ask:
  - what does the mechanism achieve?
  - do we need confidentiality, integrity or availability of this data?
  - who will generate the keys and how?
  - who will store/ have access to the keys?
  - what if we lose keys?
  - will it interfere with other security measures?
  - will it introduce new vulnerabilities?
  - what if it breaks or is broken?

Design Hierarchy

- What are we trying to do?
- How?
- With what?



You (your organization) need to

Step 1: security requirement analysis
- identify assets and their values
- identify vulnerabilities, threats and risk priorities
- identify legal and contractual requirements

Step 2: work out a high-level security policy
: requirements are to be abstracted first into a high-level security policy(what security means and what needs to be protected or enforced)

Step 3: security policy documentation
: as a reference for anyone involved in implementation

Step 4: detailed design and implementation

Security vs Dependability

- Dependability = safety+ security
- Safety and security are often strongly correlated in practice
- But malice is different from error!
  - safety: “Bob will be able to read this file”
    (free from accidents)
  - security: “Trudy won’t be able to read this file”
    (free from damage by attacks)

A Framework

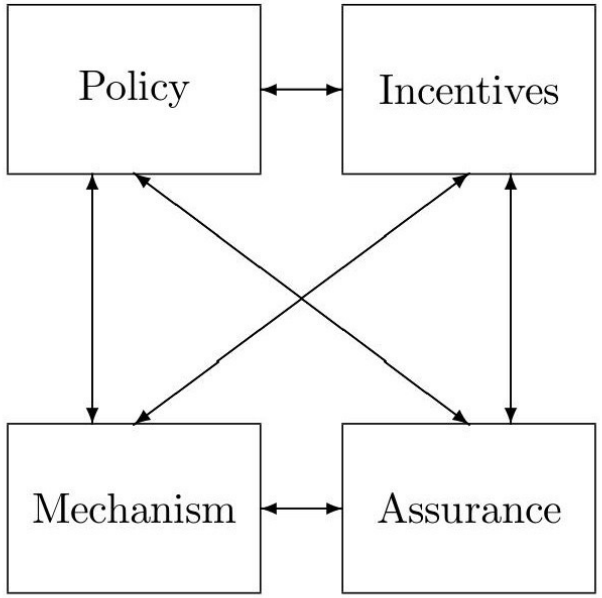
- 4 things to come together to build really dependable systems:
  - policy: what you're supposed to achieve
  - mechanism: used to implement the policy

    (ciphers, access controls, hardware tamper-resistance, …)
  - assurance: the amount of reliance you can place on each

    mechanism and how well they work together

A Framework

- incentive: motive that people have to do their job properly and that attackers have to try to defeat your policy



A Framework

- Example: make airport security systems dependable
  - assumption: hijacker's trying to getting knives and explosives on board
  - policy failure of not screening knives
  - weak mechanism of keeping out explosives not containing nitrogen
  - poor assurance: less than half of all real weapons are spotted and confiscated
  - incentives on the decision makers favor visual controls (eg. passenger screening) over effective ones

A Framework

- Lesson learned:

  We, security engineers, need to have a wide understanding of

  - what has gone wrong over time with various systems
  - what sort of attacks have worked
  - what their consequences were
  - how they were stopped

Case Study 1: a bank

- operations on a core bookkeeping system
  - main threat: bank's own staff
  - defense by bookkeeping procedures
    - : each debit must be matched by a credit
      (so money only be moved within a bank)
    - : large transfers need two to authorize them
    - : alarms on unusual volumes or patterns of transactions
    - : staff required to take regular vacations
- via ATM
  - threat: card and PIN base authentication

Case Study 1: a bank

- via bank's website and mobile apps
  - threat: social engineering, phishing, SIM swapping, …
- via high volume messaging system(move large sums between banks)
  - threat: attacks on the system to steal many millions
- via bank's branches
  - threat: break in and rob bank's branches

Case Study 2: a military base

- encrypted military communications
  - threat: cryptanalysis, locate and attack the transmitter by detecting encrypted traffic
- jamming enemy radars while preventing the enemy from jamming yours: spoofing and service-denial attack
- developed multi-level security system: Top Secret, Secret, Confidential, Restricted
- developed authentication schemes (for protecting nuclear weapons): biometrics(iris pattern), …

Case Study 3: a hospital

- medical devices' safety and security: eg. infusion pumps for drip-feeding patients with drugs (w/ software + RF communications)
- protecting patient record system
  - how to *effectively* implement, say, "nurses can see the records of any patient who has been cared for in their department at any time during the previous 90 days"
- encrypting patient names is not enough: eg. "show me all males born in 1953 who were treated for atrial fibrillation on Oct 19th 2003"

Case Study 3: a hospital

- failures can be big and fatal
  - eg. ransomware made the patient record system encrypted (and not infected backup is a few days old)
  - eg. worm penetrated into the hospital and made electricity and water supplies fail

Case Study 4: the home

- may have lots(?) of security-sensitive systems
  - PC/phones running banking applications
  - insulin pump communicating with a docking station at your bedside
  - home burglar alarm sending encrypted 'all's well' signals
  - IoT system connecting cooling/heating controllers, home security devices, …
  - Alexa and Google Home listening to what people say
- cars w/ electronic immobilizer

Definitions

- A *system* can be:
  - a product or component (protocol, smartphone, smartcard, laptop, server, …)
  - some products plus OS, communications and other infrastructure
  - the above plus applications(banking app, health app, media player, browser, …)
  - the above plus IT staff
  - the above plus internal users and management
  - the above plus customers and external users
- Common failing: policy drawn too narrowly

Definitions

- A *subject* is a physical person
- A *person* can also be a legal person(company, government)
- A *principal* is an entity that participates in a security system and can be
  - a person
  - equipment (laptop, phone, smartcard, card reader, …)
  - communication channel(port number, crypto key, …)
  - a compound of other principals(Alice or Bob, Bob acting as Alice's manager, Alice acting for Bob in his absence, …)
- The level of precision is variable – sometimes you need to distinguish 'Bob's smartcard representing Bob who's standing in for Alice' from 'Bob using Alice's card in her absence'. Sometimes you don't

Definitions

- An *identity* is a correspondence between the names of two principals to note they refer to the same person or equipment

  eg. Bob in "Alice acting as Bob's manager' is the same as the Bob in "Bob as branch manager signing a bank document jointly with David" ?

- a *group* means a set of principals
- a *role* is a set of functions assumed by different persons in succession (such as 'the officer of the watch on the USS Nimitz')

Definitions

- *Trust* is the hard one! It has several meanings:
  1. a warm fuzzy feeling
  2. a trusted system or component is one that can break my security policy
  3. a trusted system is one I can insure
  4. a trusted system won't get me fired when it breaks
- I'm going to use the NSA definition – number 2 above – by default. E.g. an NSA man selling key material to the Chinese is *trusted* but not *trustworthy* (assuming his action unauthorized)

Definitions

- *Secrecy* is a technical term – mechanisms limiting the number of principals who can access information, such as cryptography or computer access controls
- *Privacy* means control of your own secrets is the ability and/or right to protect your personal info and to prevent invasions of your personal space
- *Confidentiality* is an obligation to protect someone else's secrets if you know them
- Thus your medical privacy is protected by your doctors' obligation of confidentiality

Definitions

- *Authenticity* in security protocols means integrity plus freshness: "you are speaking to a genuine principal, not a replay of previous message"

- *Integrity* means the state or guarantee of "(target) not having been altered/mangled/faked/manipulated since the last authorized modification"

Definitions

- a *hack* is something a system's rules permit, but which was unanticipated and unwanted by its designers (definition by Bruce Schneier)

  - you may try to hack a cryptosystem by

    1. finding a mathematical weakness in the encryption algorithm, or
    2. measuring the power drawn by a device that implements it to work out the crypto key
    3. deceiving the device's custodian into using it when they shouldn't

Definitions

- *Anonymity* is about restricting access to metadata. It has various flavors, from not being able to identify subjects to not being able to link their actions

Definitions

- A *security policy* is a succinct statement of protection strategy (eg. "in each transaction, sums of credits an debits are equal, and all transactions over $1,000,000 must be authorized by two managers")
- A *security target* is a detailed statement which sets out the means(encryption, access control, …) by which a security policy will be implemented in a particular product and be used as the yardstick to evaluate if the engineers have done a proper job
- A *protection profile* is like a security target, except written in a sufficiently device-independent way to allow comparative evaluations among different products

Definitions

- A *vulnerability* is a property of a system or its environment which, in conjunction with an internal or external *threat*, can lead to a *security failure*, which is a breach of the system's security policy

C.I.A

- CIA == Confidentiality, Integrity, and Availability

- **Confidentiality:** preventing unauthorized *reading* of information
- **Integrity:** detecting unauthorized *writing* of information
- **Availability:** ensuring data is available in a timely manner when needed

Beyond C.I.A

- CIA are only beginning of the Inf Security
- Case 1: when Bob logs on his computer
  - How does Bob's computer know that "Bob" is really Bob and not Trudy?
- Bob's password must be verified
  - This requires some clever **cryptography**
- What are security concerns of pwds?
- Are there alternatives to passwords?

Beyond C.I.A

- Case2: when Bob logs into AOB
  - how does AOB know that "Bob" is really Bob?
- As before, Bob's password is verified
- Unlike standalone computer case, network security issues arise
- What are network security concerns?
  - **Protocols** are critically important
  - Crypto also important in protocols

Beyond C.I.A

- Once Bob is *authenticated* by AOB, then AOB must restrict actions of Bob
  - Bob can't view Charlie's account info
  - Bob can't install new software, etc.
- Enforcing these restrictions is known as authorization
- **Access control** includes both authentication and authorization

Beyond C.I.A

- Cryptography, protocols, and access control are implemented in **software**
- What are security issues of software?
  - Most software is complex and buggy
  - Software flaws lead to security flaws
  - How to reduce flaws in software development?

Beyond C.I.A

- Some software is intentionally evil
  - Malware: computer viruses, worms, etc.
- How do the malwares work?
- What can Alice and Bob do to protect themselves from malware?
- What can Trudy do to make malware more “effective”?

Think Like Hacker

- In the past, no respectable sources talked about “hacking” in detail
- It was argued that such info would help hackers
- Very recently, this has changed
  - Books on network hacking, how to write evil software, how to hack software, etc.
- Good guys must think like bad guys!
- A police detective
  - Must study and understand criminals

Q & A

aiclasshongik@gmail.com