Class Overview

Yajin Zhou (http://yajin.org)

Zhejiang University

Learning Objectives

- Understand common vulnerabilities and attacks
  - Buffer overflow, ROP…
- Understand program analysis methods/tools
- Learn how to analyze programs and write exploits
- Know how to write safe code

Prerequisites

- Operating systems
- C and assembly language
- Computer systems

Instructor

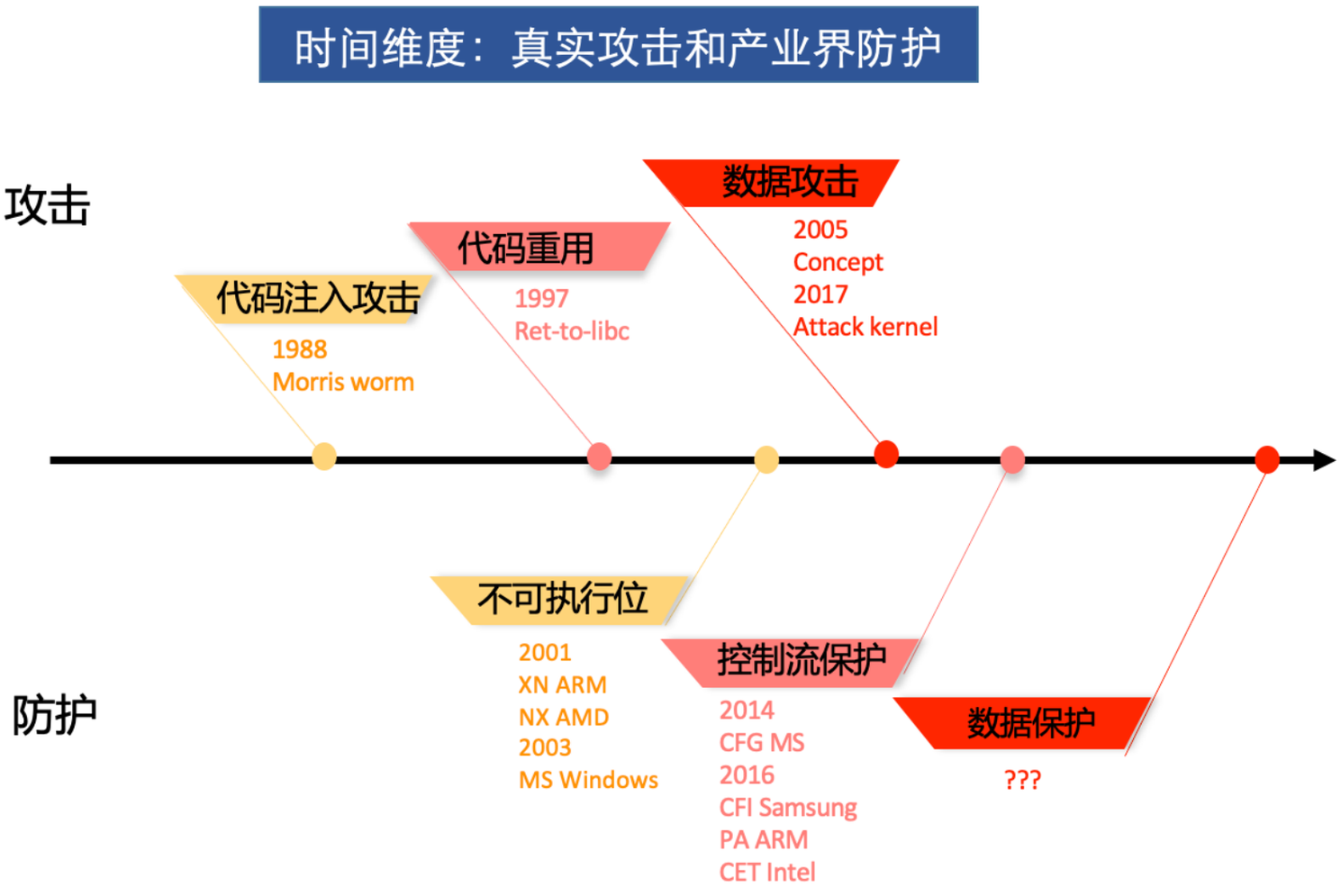
- Yajin Zhou (yajin_zhou@zju.edu.cn)
- Research
  - software security, operating systems security, hardware-assisted security and confidential computing.
  - Emerging threats: security of smart contracts, decentralized finance (DeFi) security, and underground economy

Research Projects

- Interested in security and system?
  - Want to build system and make system more secure
  - Want to hack a system
- Ping me if you are interested

Attack & Defense

- 攻击演化
  - 攻击难度指数级增加
    - 复杂性指数级增加
    - 隐蔽性在增加
  - 控制能力缩小
    - 数据攻击依然能root内核
- 防护演化
  - 软件到硬件
  - 学术界原型到产业界实用方案
  - 有滞后性



Introduction to Software Security

Yajin Zhou (http://yajin.org)

Zhejiang University

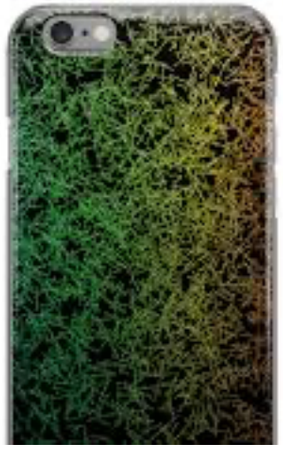
Credits: Part of slides are from cse443 of PSU (Trent Jaeger), CS-412 of EPFL(mathias payer), CMPSC 447 of PSU(Gang Tan)

Security vs safety

NTNU definition (Skavland Idsø and Mejdell Jakobsen, 2000):

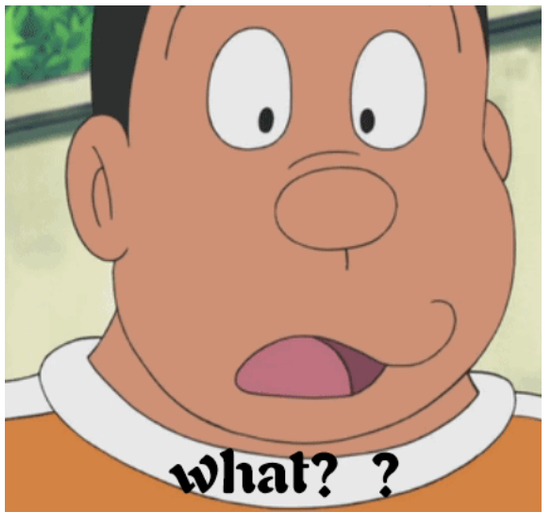
Safety is protection against random incidents. Random incidents are unwanted incidents that happen as a result of one or more coincidences.

Security is protection against intended incidents. Wanted incidents happen due to a result of deliberate and planned act.



Why we need a course on software security

- Software plays an important role
  - But is also a major source of security problems
- Software security does not get much attention
  - In other security courses or
  - In programming courses



Scope

- Software Security is the area of Computer Science that focuses on **(i) testing, (ii) evaluating, (iii) improving, (iv) enforcing, and (v) proving the security of software.**

- Learn to identify common security threats, risks, and attack vectors for software systems

- Assess current security best practices and defense mechanisms for current software systems

- Design and evaluate secure software. Have fun!

Software **In**security

Morris Worm


The Morris Internet Worm source code
This disk contains the complete source code of the Morris Internet worm program. This tiny, 99-line program brought large pieces of the Internet to a standstill on November 2nd, 1988.
The worm was the first of many intrusive programs that use the Internet to spread.
Computer History Museum
Internet Worm - Source code
X1294.96A-D

Morris Worm

- First worm: Nov 1988, infected more than 10% of Inernet
  - **Buffer overflow** in fingerd, injected shellcode and commands
  - Debug mode in sendmail **to execute arbitrary commands**
    - **Dictionary attack** with frequently used usernames/passwords
- **Buggy worm:** the routine that detected if a system was already infected was faulty and the worm kept re-infecting the same machines until they died.
- Link: [Examining the Morris Worm Source Code - Malware Series - 0x02]

Microsoft Zune Crash

- December 31st 2008, owners of Microsoft's Zune MP3 player found that their devices were freezing at start-up.
- *From what I can tell it looks like every Zune 30 on the planet has suddenly crashed. Is this a virus? A glitch? A time bomb? A disgruntled Microsoft employee? Planned obsolescence to make us buy a new one? Or just a terrorist plot to drive the free world crazy?*


```
year = ORIGINYEAR; /* = 1980 */
while (days > 365) {
  if (IsLeapYear(year)) {
    if (days > 366) { days -= 366; year += 1; }
  } else { days -= 365; year += 1; }
}
```

What's the problem here?

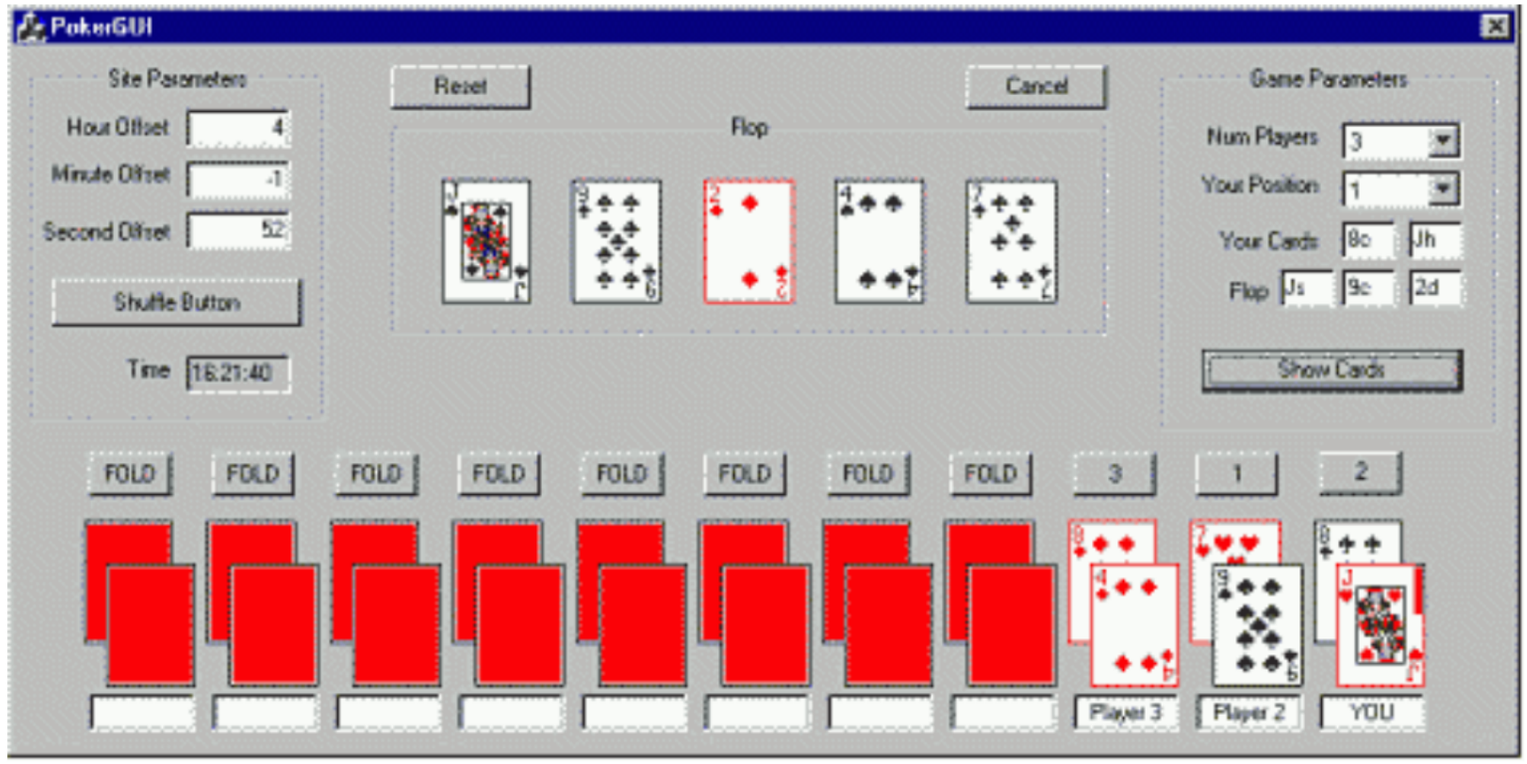
Programming Bug: Poker Site Flaw

- Web site where users can play poker over the Internet

Picture taken from Cigital Press Release

Programming Bug: Poker Site Flaw

- Security engineers wrote a program to “predict” cards of opponents:



- Exploited flaw: bad random number generation in shuffling cards

People are making same mistakes

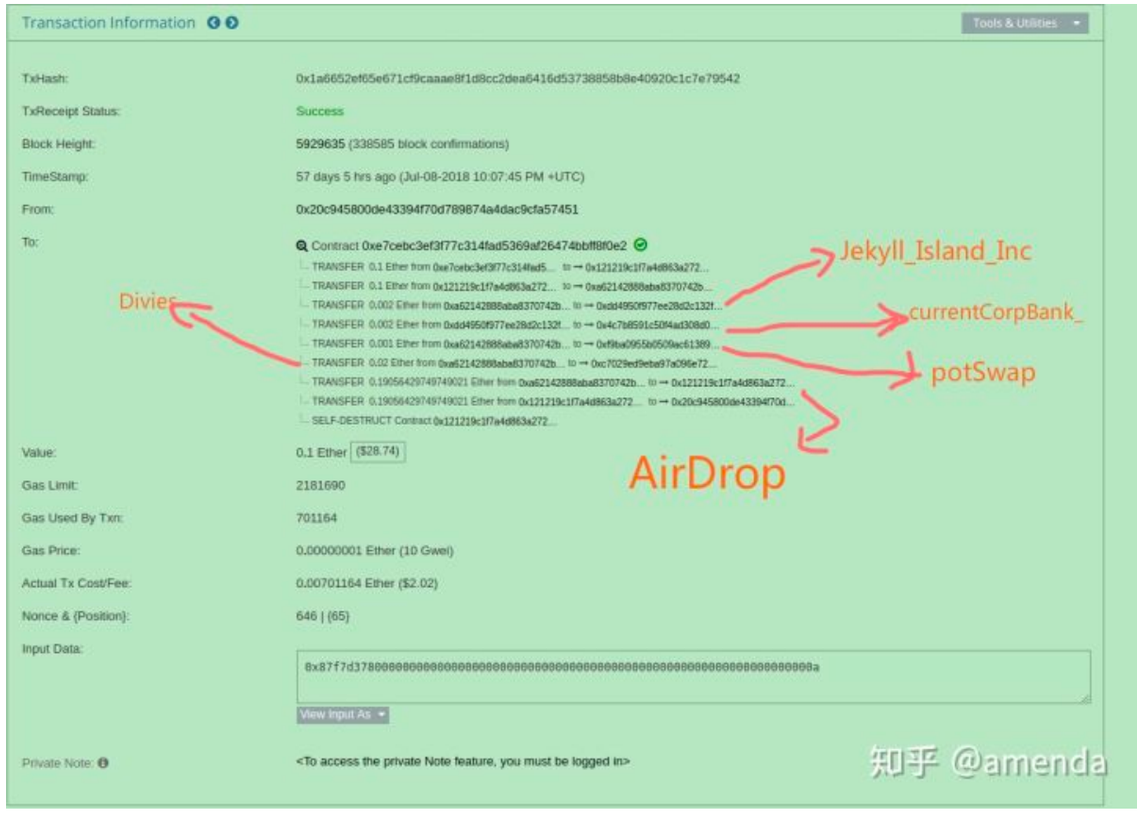
2018年7月6号，一款名叫FoMo3D的智能合约(0xA62142888ABa8370742bE823c1782D17A0389Da1)悄然上线，在之后的一个月内，由于其极具吸引力的规则，迅速火爆起来，第一轮游戏一直持续到8月22号才结束，吸金上亿，我们先对FoMo3D进行一个简单了解：

- FoMo3D是一款资金盘游戏，每轮游戏初始24个小时
- 玩家通过购买seed加入游戏， 每有一位玩家购买，本轮的游戏时间会延长30秒（可见第一轮游戏的火爆程度）
- 每次购买seed会增加seed的售价
- 每次购买seed的金额会分为四个部分，一部分要给先入场的玩家分红，分红比例取决于玩家各自购买的seed数量占比，一部分流入最终奖池，一部分流入空投奖池，一部分给介绍人(上线)
- 流入空投奖池的金额占每次购买的1%，给上线的占10%，流入最终奖池和分红的占比取决于选择的战队，战队系统就不多介绍了，与本文关系不大
- 每次购买花费超过0.1ETH有一定几率获得空投奖励
- 一轮游戏结束后，最终奖池的48%由最后一个购买seed的大赢家获得，剩下的52%一部分流入下一个奖池，一部分给所有玩家分红，比例取决于大赢家所属战队

People are making same mistakes

- However sources to generate random numbers are not truly random
- Attackers can predict whether they can get the bonus. If so, they will buy the seed to bid

https://zhuanlan.zhihu.com/p/44274223

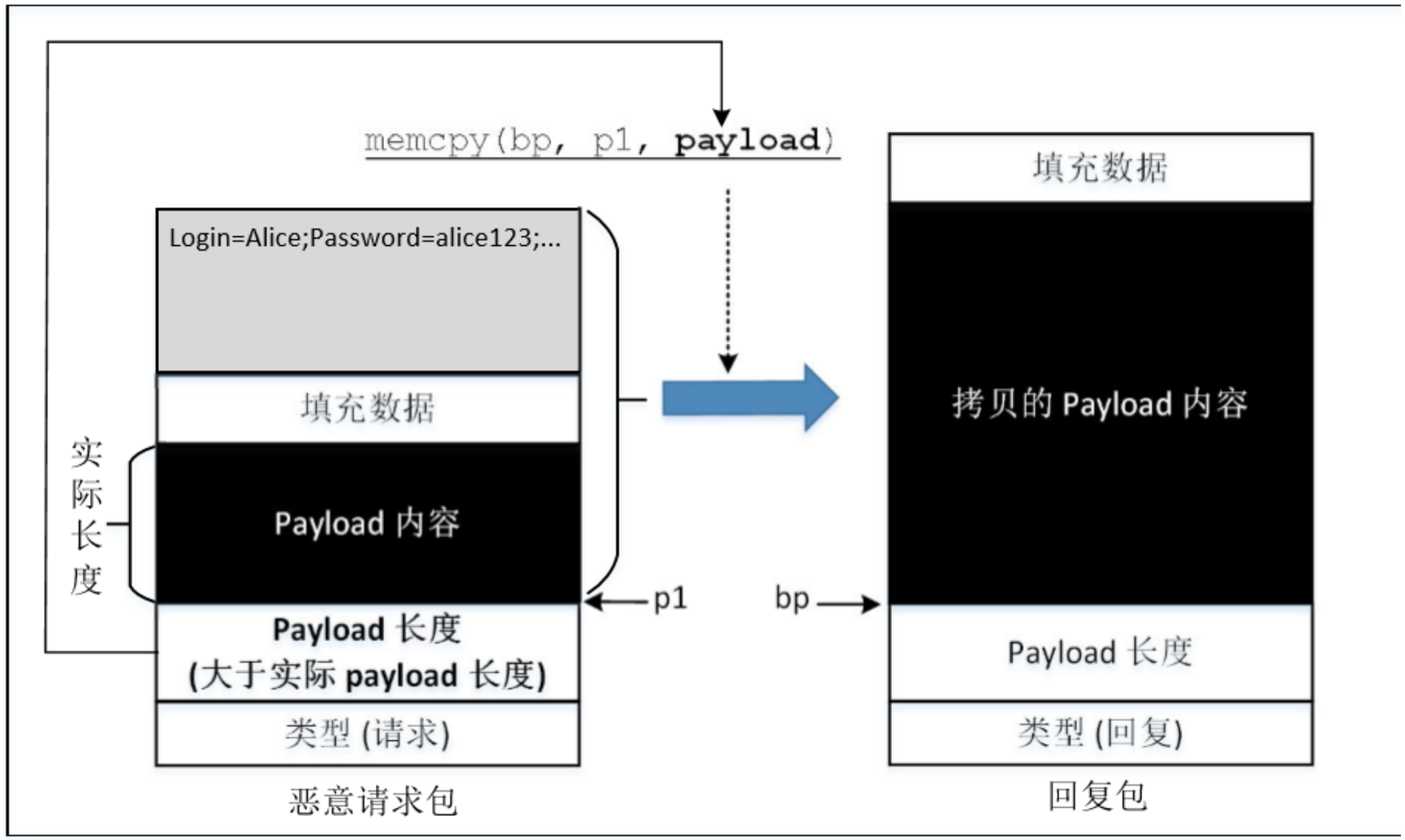


OpenSSL Heartbleeding

- CVE-2014-0160 is the official reference to this bug. CVE (Common Vulnerabilities and Exposures) is the Standard for Information Security Vulnerability Names maintained by MITRE. Due to co-incident discovery a duplicate CVE, CVE-2014-0346, which was assigned to us, should not be used, since others independently went public with the CVE-2014-0160 identifier.

- http://heartbleed.com/

OpenSSL Heartbleeding



Complexity

- Why Can’t Programmers be More Careful?
- Complexity
  - Software becomes more and more complicated.
  - Size is measured in terms of millions lines of code

    Low-level languages (C/C++) trade type safety and memory safety for performance

    Google Chrome: 76 MLoC

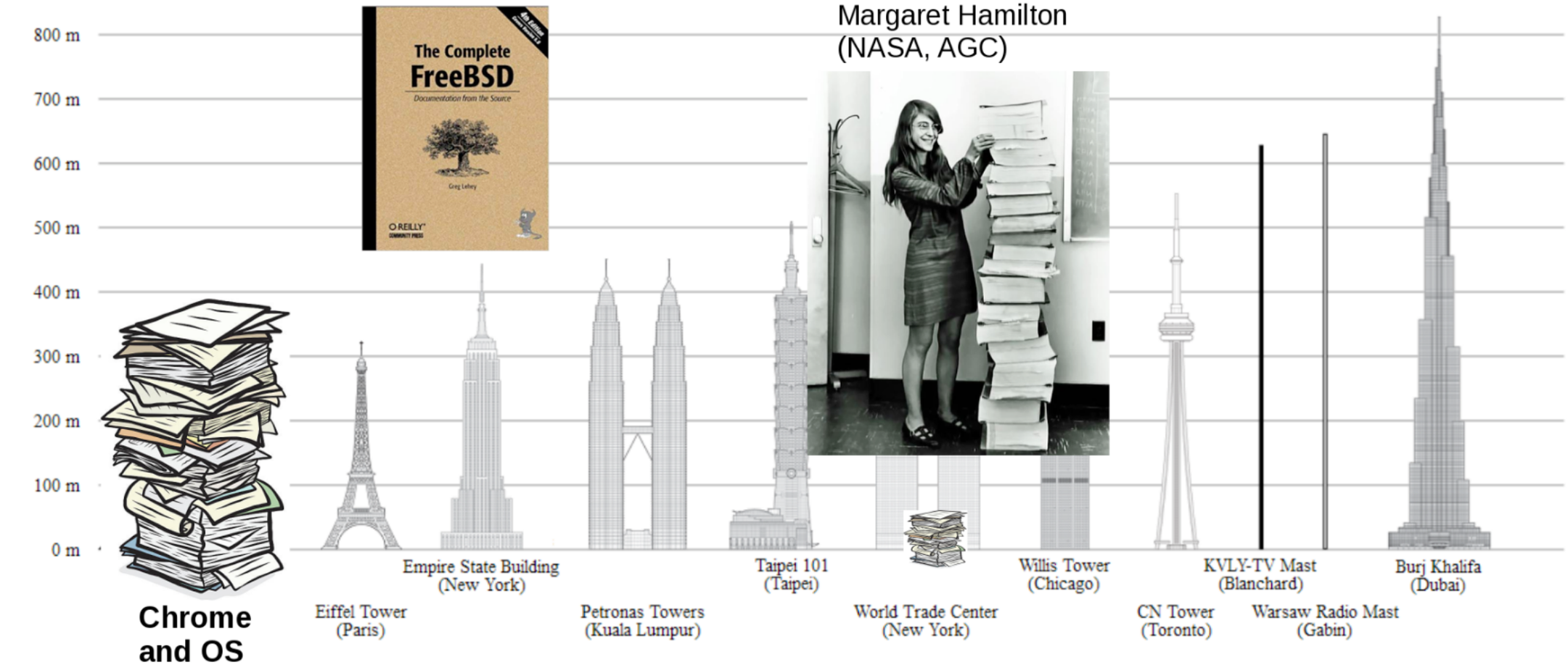
    Gnome: 9 MLoC

    Xorg: 1 MLoC

    glibc: 2 MLoC

    Linux kernel: 17 MLoC

Complexity



~100 mLoC, 27 lines/page, 0.1mm/page equals roughly 370m

Connectivity

- Connectivity
  - The Internet makes it possible for attackers to
  - Exploit software remotely
  - IoT ear makes this even worse

Understanding the Mirai Botnet

Manos Antonakakis[◇] Tim April[‡] Michael Bailey[†] Matthew Bernhard[◁] Elie Bursztein[○]
Jaime Cochran[▷] Zakir Durumeric[◁] J. Alex Halderman[◁] Luca Invernizzi[○]
Michalis Kallitsis[§] Deepak Kumar[†] Chaz Lever[◇] Zane Ma[†*] Joshua Mason[†]
Damian Menscher[○] Chad Seaman[‡] Nick Sullivan[▷] Kurt Thomas[○] Yi Zhou[†]

[‡]*Akamai Technologies* [▷]*Cloudflare* [◇]*Georgia Institute of Technology* [○]*Google*
[§]*Merit Network* [†]*University of Illinois Urbana-Champaign* [◁]*University of Michigan*

https://www.usenix.org/system/files/conference/usenixsecurity17/sec17-antonakakis.pdf

Extensibility

- Extensibility
- Software systems are not closed
  - App market: allow users to extend the functionality of their phones
- However
  - We don’t know who wrote those apps?
  - What if an app steal our credit card info or track our locations?
- Like connectivity, hackers also like extensible systems
  - Giving them an opportunity to inject malicious code

Why software enginnerring cannot save us

- Software engineering aims for
  - **Dependability**: producing fault-free software
  - **Productivity**: deliver on time, within budget
  - **Usability**: satisfy a client's needs
  - **Maintainability**: extensible when needs change
- Software engineering combines aspects of PL, networking, project management, economics, etc.
- Security is secondary and **often limited to testing**.

Topics

- Vulnerabilities and Attacks
  - Buffer overflow, Return2libc/ROP, Format String Vulnerabilities
- Code analysis
  - Static analysis, taint analysis, symbolic/eoncolic execution, fuzzing
- Advanced topics
  - CFI/SFI
  - Hardware-assisted protection
  - AEG

Course Material

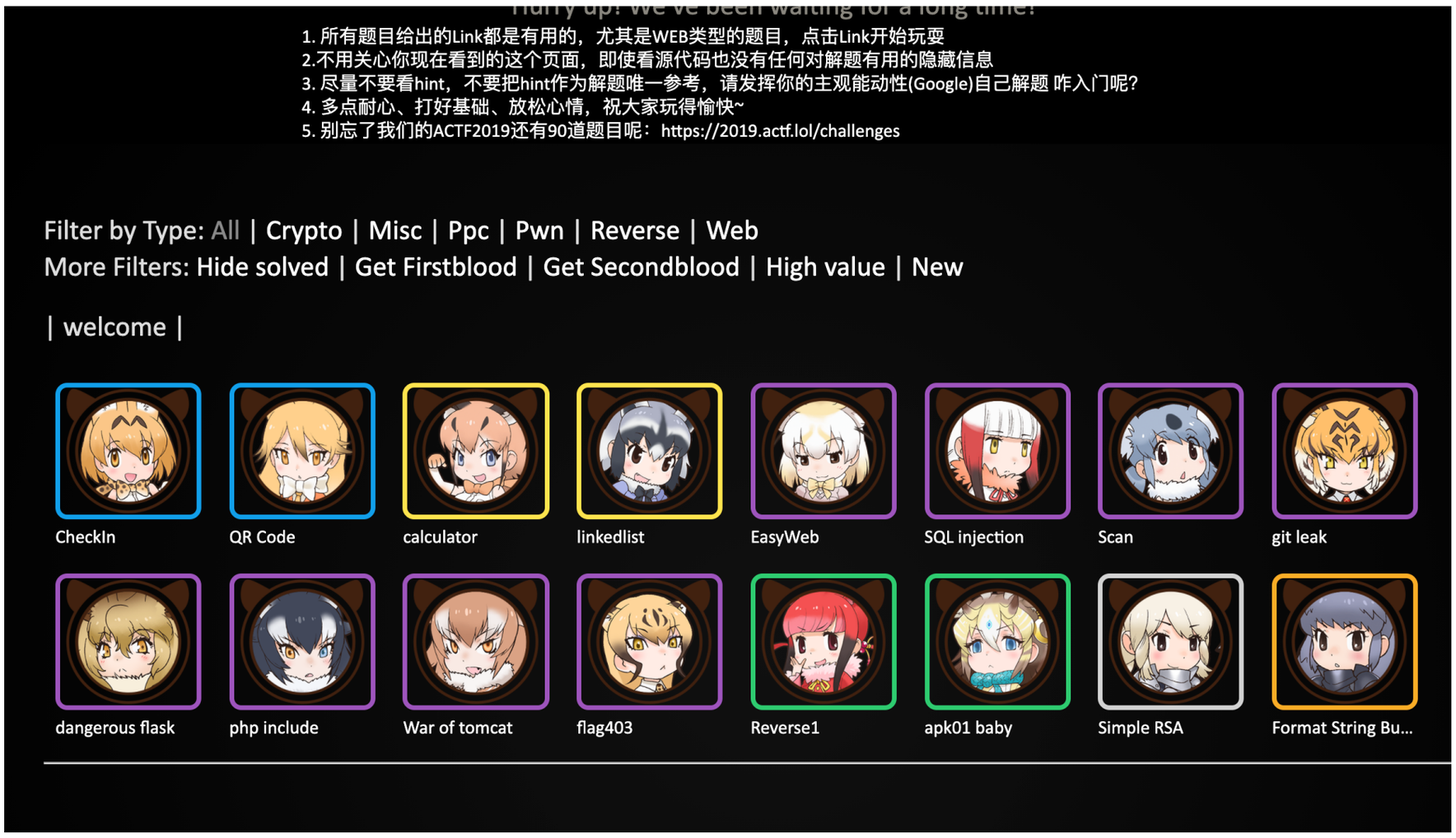
- Lecture notes (posted at the class website). Please check frequently
- http://course.zju.edu.cn
  - Ask questions
- TAs
  - 马麟
  - 周多明
  - 卜誉杰

Course Material

- Reference
  - Software Security: Principles, Policies, and Protection
  - CMPSC 447 Software Security – PSU
  - Software Security - EPFL

Course Material

- School bus: https://zjusec.com/play
- https://2019.actf.lol/challenges



Grading

- Homework - 60%, Final project - 40%
- Late submissions are accepted after the deadline
  - a **10% penalty** will be applied for each day of late submission
  - Disputes of grade MUST be resolved within one week of receiving it
- Homework assignments
  - Some written assignments and some projects (labs)

Academic Integrity

- Do not copy code from others!

Ethical Issue

- This class may contain technologies whose abuse may infringe on rights of others! **Do not undertake any action which could be perceived as technology misuse under any circumstances unless you have received explicit permissions.**