



1 软件安全概述

网络空间安全学院 慕冬亮
Email : dzm91@hust.edu.cn

软件安全：一门对抗性学科

攻击者



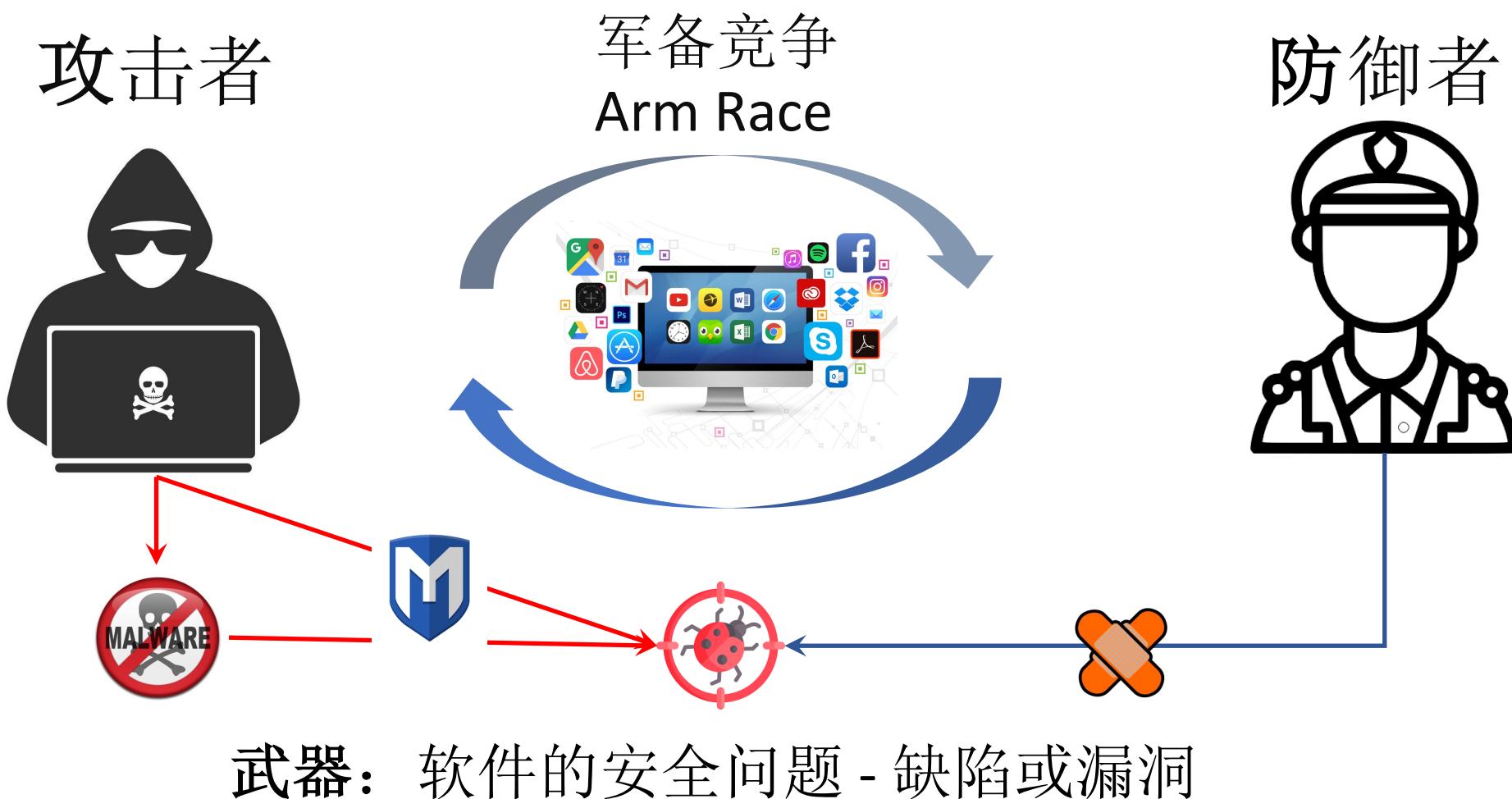
军备竞争
Arm Race



防御者



软件安全：一门对抗性学科



软件的安全问题

- ❖ 课程内容简介
- ❖ 最新安全大数据
- ❖ 任何软件都是不安全的
- ❖ 软件不安全性的几种表现
- ❖ 软件不安全的原因
- ❖ 怎么考虑软件安全问题？

软件的安全问题

- ❖ 课程内容简介
- ❖ 最新安全大数据
- ❖ 任何软件都是不安全的
- ❖ 软件不安全性的几种表现
- ❖ 软件不安全的原因
- ❖ 怎么考虑软件安全问题？

软件系统简介

- ❖ 软件系统，是计算机应用的重要组成部分。尽管软件开发者竭尽努力，但软件中不可避免地包含软件缺陷或软件漏洞。当这些缺陷被**无意或有意**触发后，软件会崩溃或非正常退出，将会给用户带来巨大的损失。此外，恶意攻击者还会利用缺陷攻陷软件系统，从而获取有价值的数据信息，用于牟取利益。
- ❖ 然而，面对软件缺陷带来的巨大损失，现实世界对软件开发者提出了更高的要求，要求程序员能够编写出错误更加少的程序，并且能够及时修复软件出现的突发问题，切实为软件用户服务。



火星气候探测者号“失联”



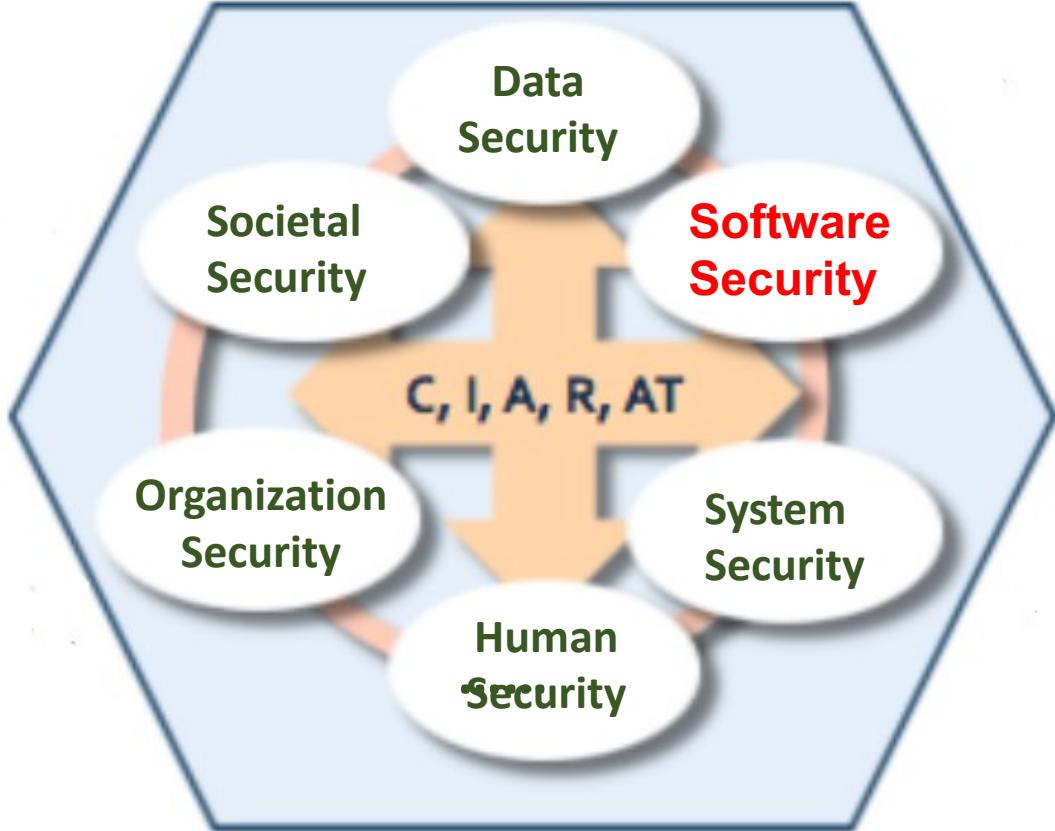
莫里斯蠕虫

课程内容简介

- ❖ 软件安全中的最大挑战 – 软件漏洞，其产生原因，分类，危害？怎么挖掘、分析与利用软件漏洞？如何利用安全编程及安全机制来抵御软件漏洞？
- ❖ 软件中的“异类” – 恶意软件现在越来越影响到人们的日常生活，它们如何工作？如何分析以及防御这些恶意软件？
- ❖ 本课程主要就是针对以上这些问题进行介绍，了解软件安全的核心技术。同时，安全编程是软件质量的重要保证，在软件开发和程序设计中具有重要地位。



课程内容简介



- ◆ CSEC2017模型有八个知识领域：**软件安全**，系统安全，数据安全，组件安全，连接安全，人员安全，组织安全以及社会安全。
- ◆ CSEC2017思想模型包括学科六个内涵属性：C-机密性，I-完整性，A-可用性，R-风险，AT-对抗，ST-系统性思考

简介

- ❖ **1. 软件安全概述** 软件安全威胁、概念；软件安全所涉及的技术范畴以及软件安全关键技术与措施分析
- ❖ **2. 软件安全技术基础** x86汇编语言基础知识、系统引导与控制权、操作系统虚拟内存、ELF可执行文件以及软件逆向工具
- ❖ **3. 软件缺陷与漏洞机理基础** 安全攻击事件、漏洞分类及标准、漏洞生命周期、漏洞影响、产生原因、利用方式及典型漏洞
- ❖ **4. 内存安全与漏洞分析** 堆栈、函数调用原理、栈溢出、堆溢出、整数溢出、格式化溢出、软硬件漏洞防御等
- ❖ **5. 漏洞利用与发现** 漏洞利用、Shellcode、利用实战、平台、框架与工具，代码审查，静/动态挖掘及模糊测试等
- ❖ **6. 构建安全的软件** 威胁建模，安全代码编写，漏洞响应和维护，SDL

简介

❖ 7. 恶意代码及其分类

❖ 8. Windows PE 病毒

❖ 9. 宏病毒与脚本病毒

❖ 10. 网络蠕虫

❖ 11. 木马与后门

❖ 12. 恶意代码防范技术

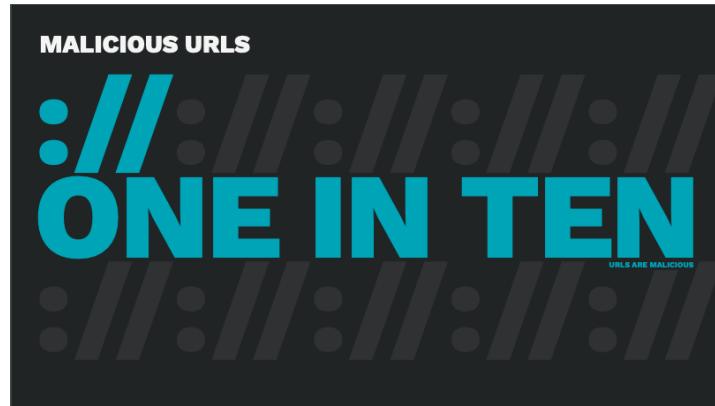
软件的安全问题

- ❖ 课程内容简介
- ❖ 最新安全大数据
- ❖ 任何软件都是不安全的
- ❖ 软件不安全性的几种表现
- ❖ 软件不安全的原因
- ❖ 怎么考虑软件安全问题？

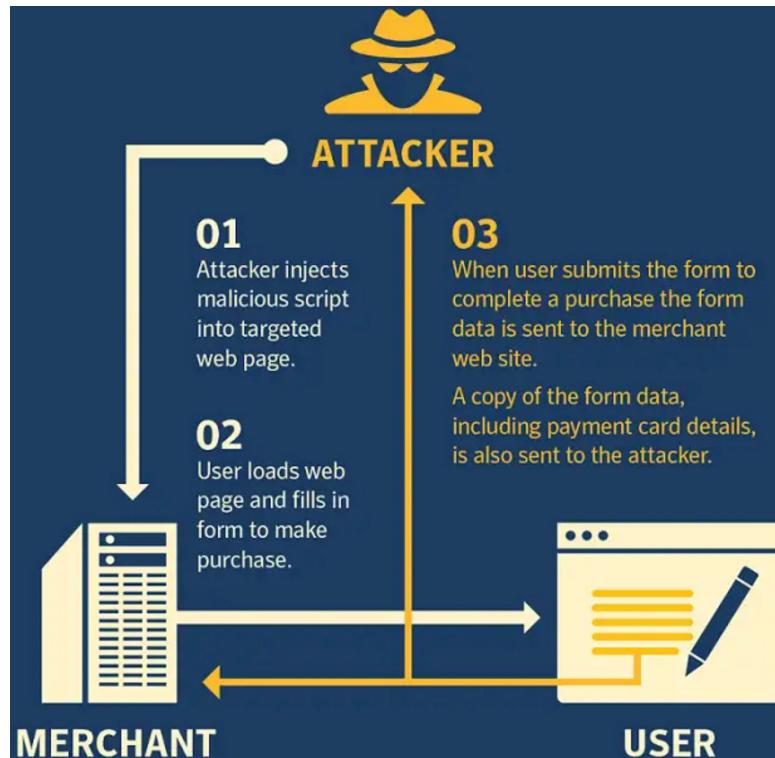
最新软件安全态势

- 先来了解国内外最新态势
- The Newest Malware BigData

赛门铁克2019互联网安全威胁报告(1)



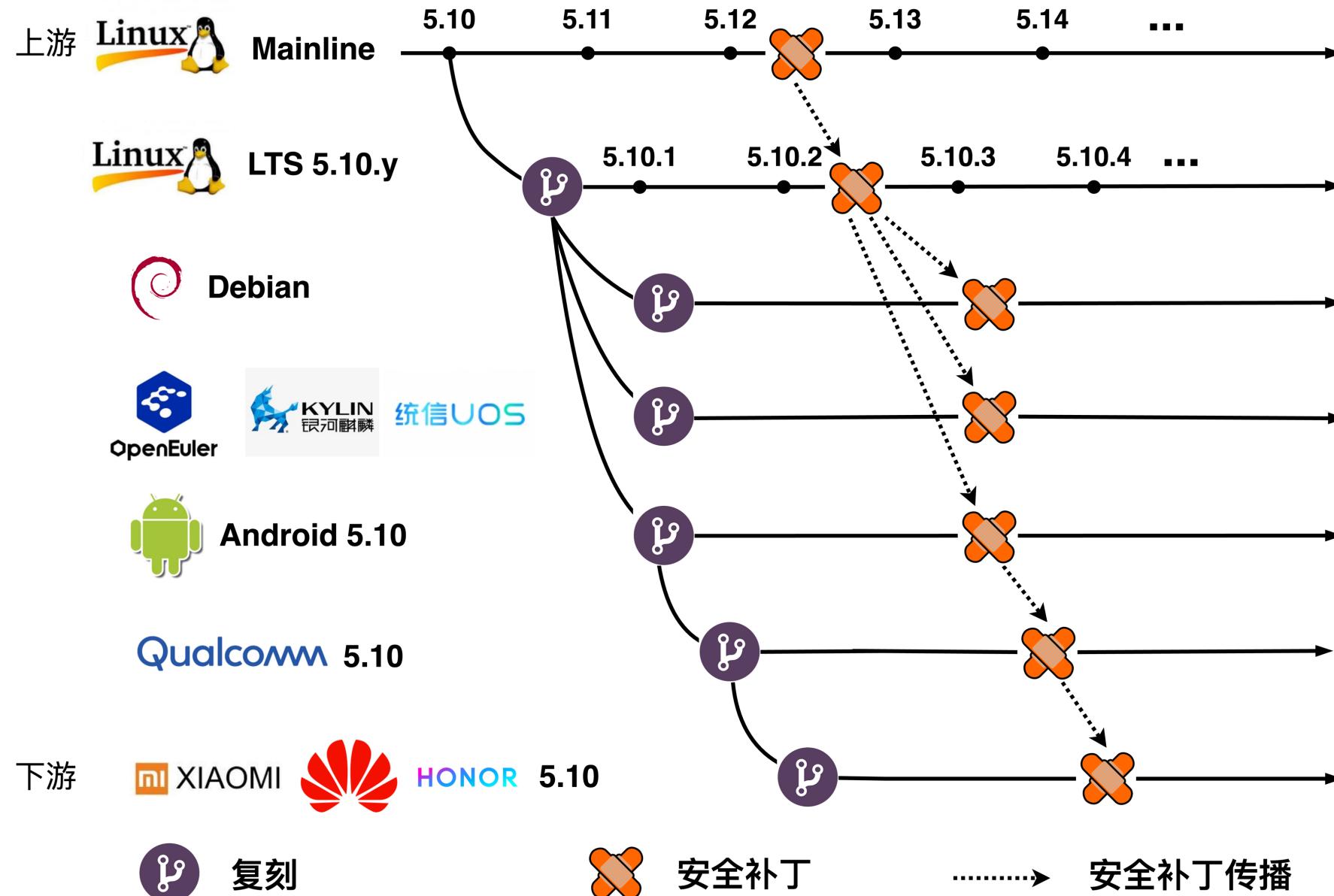
Formjacking Attack



Screenshot of a merchant website showing a form for a purchase. The payment details section is highlighted with a red box:

1 item in cart
Delivery / Pick up
Pick Up/Delivery Time and Special Instructions
Delivery time: ASAP (about 30 mins) Choose Date and Time
Today asap
Delivery & Allergy Instructions: e.g. Allergies or instructions for the driver
Add Payment Details
Contact Details: First name, Last name, Mobile, Email
Payment Details: Credit Card, PayPal
Credit card number: 16-digit on front of card
Expiry date: Month, Year
CVV number*: 3 or 4 digits
Total: (GST inclusive) \$30.00
Place Order Now

赛门铁克2019互联网安全威胁报告(2)



2020年美国联邦政府数据泄露事件

- SolarWinds 网络攻击活动
- 2020年，一个由他国政府支持的组织发动了一场大规模网络攻击。全球包括美国各级政府部门、北约、英国政府、欧洲议会、微软等至少200个政府单位、组织或公司受到影响，其中一些组织的数据可能也遭到了泄露。由于此次网络攻击和数据泄露事件持续时间久，目标知名度高、敏感性强，多家媒体将其列为美国遭受过的最严重的网络安全事件。
- 微软漏洞、SolarWinds漏洞利用、VMware漏洞利用

赛门铁克2019互联网安全威胁报告 (3)



**ROUTERS AND CONNECTED CAMERAS
WERE THE MAIN SOURCE OF IOT ATTACKS
ACCOUNTING FOR OVER
90 PERCENT
OF ACTIVITY.**

IOT DEVICES EXPERIENCE AN AVERAGE OF 5,200 ATTACKS PER MONTH



TOP SOURCE COUNTRIES FOR IOT ATTACKS (YEAR)

COUNTRY	PERCENT
China	24.0
USA	10.1
Brazil	9.8
Russia	5.7
Mexico	4.0
Japan	3.7
Vietnam	3.5
South Korea	3.2
Turkey	2.6
Italy	1.9

IOT Botnet: Mirai



IoT僵尸网络严重威胁网络基础设施安全

北美DNS服务商遭Mirai木马DDoS攻击的分析思考

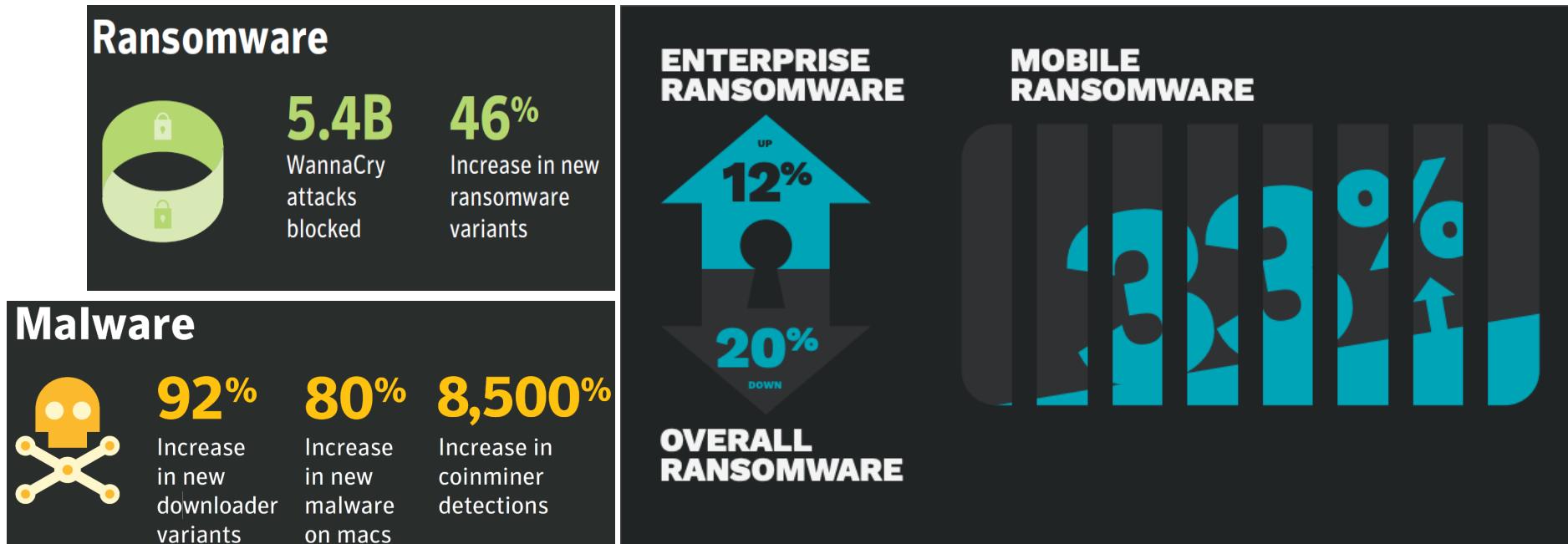
安天实验室



由Mirai僵尸网络所造成的臭名昭著的拒绝服务攻击（DDOS）是2018年中Top 3的IoT危害，约占整体攻击的16%。

Mirai 及其变种使用超过16种漏洞利用，并且还在不断地增加新的利用来增加感染新设备的成功率。

赛门铁克2019互联网安全威胁报告(4)



石油巨头遭黑客袭击，被勒索5000万美元



塑小哥

广东塑联科技有限公司 新媒体运营

塑小哥综合报道：全球最大石油生产商沙特阿美（Saudi Aramco）7月21日证实，公司的一些文件遭泄露。此前，一名网络勒索者声称获取了该公司大量数据，并要求其支付5000万美元赎金。

WannaCry Ransomware



Oops, your files have been encrypted! Chinese (tradition)

勒索病毒袭击!山东大学遭黑客攻击 学生论文数据丢失

2017-05-14 11:07

Payment will be raised on 1/4/1970 08:00:00
Time Left 00:00:00:00

Your files will be lost on 1/8/1970 08:00:00
Time Left 00:00:00:00

About bitcoin
How to buy bitcoins?
Contact Us

鲁网5月14日讯 12日，全球99个国家和地区发生超过7.5万起电脑病毒攻击事件，罪魁祸首是一个名为“想哭”(WannaCry)的勒索软件。俄罗斯、英国、乌克兰等国“中招”。中国多所高校也受到影响，许多实验室数据和毕业设计被锁。12日晚，山东大学发通知确认校内部分单位出现ONION勒索软件感染情况。

校园网被黑了?马上答辩了，论文数据全丢了

“昨晚我正改着论文，电脑突然中毒被锁，论文数据都没了”，13日，山大大四学生王猛(化名)激动地说，“快答辩了，来这么一出，整个人都不好了，光听说的就有十来个跟我一样中招的。”据了解，王猛电脑所中病毒为WannaCry蠕虫病毒。

堪比核弹的Log4j 漏洞

- 2021年12月9号深夜，技术圈经历了一场“大地震”——Apache Log4j2被曝出一个高危漏洞，危害堪比“永恒之蓝”。
- Apache Log4j2 远程代码执行，攻击者通过 jndi注入攻击的形式可以轻松远程执行任何代码。
- 该漏洞被命名为**Log4Shell**，编号**CVE-2021-44228**。由于该组件广泛应用在Java程序中，影响范围极大。



恶意代码发展态势（2019）

- 加剧：仅在19年前9个月，公开泄露事件5183起（增33.3%），数据泄露79亿条（增长112%）
- 加剧：电子邮件欺诈(BEC)连年加剧。FBI统计，18年5月上升到120亿美元。19年，达到260亿美元。
- 警惕！AI DeepFake伪造身份进行诈骗/传播虚假新闻现象值得警惕。难以检测和证伪！
- 频发：底层硬/固件漏洞频发，协议、第三方代码漏洞依然是重大隐患。谷歌、苹果、Facebook等巨头纷纷建立洞奖机制
- 警惕！与2018年相比，中美三大漏洞库公布的漏洞数量不升反降。Why? And What? 

软件安全道路依旧任重道远…

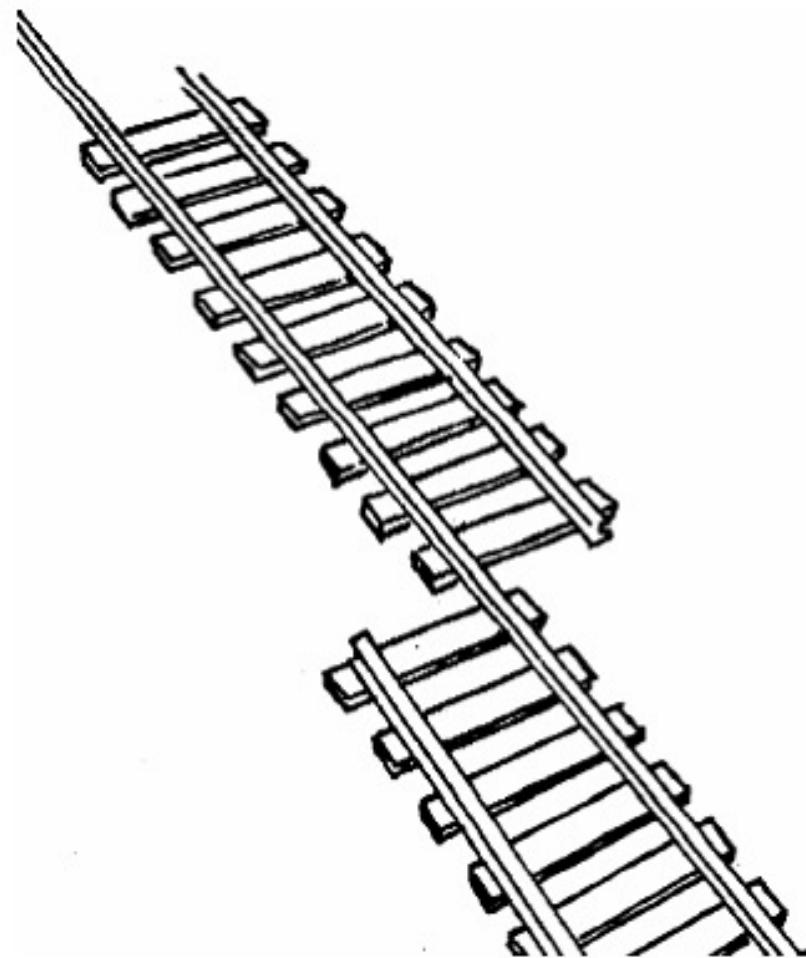
软件的安全问题

- ❖ 课程内容简介
- ❖ 最新安全大数据
- ❖ **任何软件都是不安全的**
- ❖ 软件不安全性的几种表现
- ❖ 软件不安全的原因
- ❖ 怎么考虑软件安全问题？

安全 (Safety vs Security)

- Safety
 - 自然的，物理的，相对具体的
 - 如房屋、桥梁、大坝...
- Security
 - 社会的，人为的，相对抽象的
 - 如数据、软件...

软件的安全问题？



任何软件都是不安全的

```
#define RECT2(a, b) (a * b)

int g_exam;

unsigned int example(int para, unsigned int w, unsigned int h)
{
    unsigned int temp, area;
    g_exam = para;
    area = RECT2(w + para, h);
    temp = square_exam(g_exam);
    return temp;
}
```

思考1：这个代码的后果有哪些？

任何软件都是不安全的

- 先看看软件安全问题的典型表现：

- 使用某些交易软件的过程中，某些敏感信息，如个人信息、个人卡号密码等信息被敌方获取并用于牟利；
- 访问某些网站时，服务器响应很慢，或者服务器由于访问量造成负载过大，造成突然瘫痪；
- 自己的系统中安装了具有漏洞的软件，漏洞没有解决，敌方找到漏洞并对本机进行攻击，造成系统瘫痪
- 自己花费精力完成了一幅漂亮的风景画，放到网上去，没有考虑版权，被他人随意使用却无法问责；

.....



任何软件都是不安全的

❖ 当前，软件的开发具备以下几个新的挑战：

- ❖ 软件复杂性加强
- ❖ 可扩展性要求的提高
- ❖ 开发周期日益缩短

任何软件都是不安全的

• 软件安全的挑战性

- ❖ 一方面，软件复杂了，安全问题也表现得很复杂，无法得到全面的考虑，而工程进度又迫使开发者不得不在一定时间内交付产品，代码越多漏洞和缺陷也就越来越多。
- ❖ 另一方面，软件的可扩展性要求也越来越高，系统升级和性能扩展成为很多软件必备的功能；可扩展好的系统，由于其能够用较少的成本实现功能扩充，受到开发者和用户的欢迎；但是由于针对可扩展性必须具备相应的设计，软件结构变复杂了，另外，添加新的功能，也引入了新的风险。
- ❖ 最后，移动互联时代对持续快速创新，迅速将创意转化为商品的要求，则对企业研发团队带来了更大的考验。

任何软件都是不安全的

```
/*return y=Ax*/  
int *matvec(int **A,int *x,int n)  
{  
    int *y=malloc(n, sizeof(int));  
    int i,j;  
    for (i=0;i<n;i++)  
        for (j=0;j<n;j++)  
            y[i] += A[i][j]*x[j];  
    return y;  
}
```

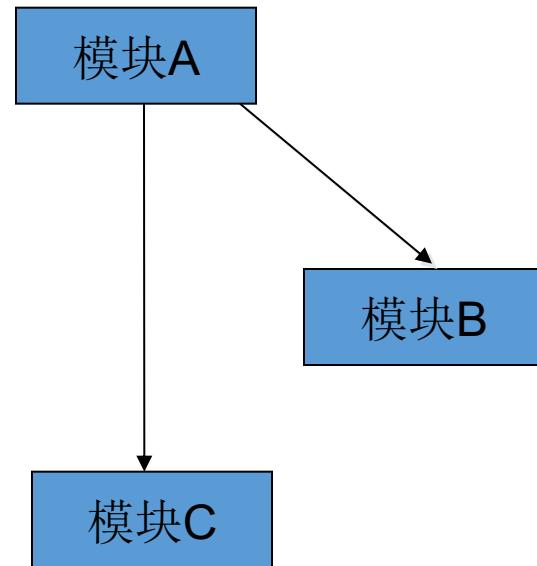
思考2：当大量的菜鸟程序员在前线，后果是什么？

任何软件都是不安全的

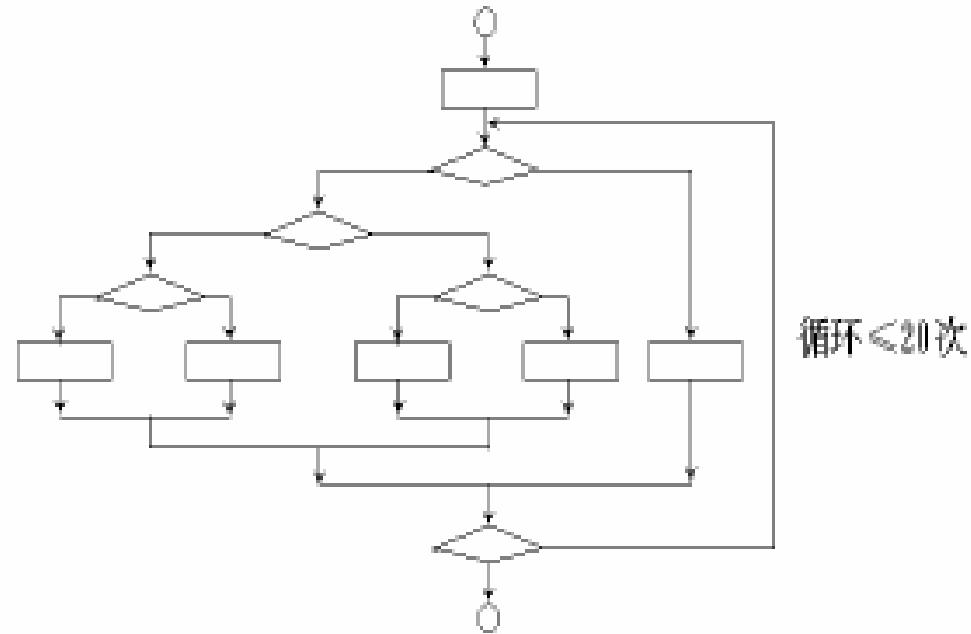
- 一般怎样解决这些安全问题?
 - ❖ 大多数人首先可以想到的方法是软件测试，通过测试来减少软件中的缺陷。
 - ❖ 但是，由于软件系统规模越来越大，软件开发的进度要求越来越高，不可能在有限的时间内考虑所有安全方面的问题，即使进行了全方位的测试，也只能对所有的测试案例进行很小范围的覆盖。

任何软件都是不安全的

- 右图所示，模块A使用模块B和模块C，以黑盒测试为例，如果模块A的输入有X种，模块B的输入有Y种，模块C的输入有Z种，理论上讲，应该对 $X \times Y \times Z$ 个组合进行全面的测试。
- 但是，由于工程进度问题，实际上在测试时不可能兼顾全面，往往只是采用了一些具有代表性的测试案例来进行测试，但这些测试案例在设计的时候又不能保证能够具有最全面的代表性。
- 如果想要将所有问题考虑到，除非进行穷举测试，而，这种穷举测试基本上是不可能完成的。



任何软件都是不安全的



举例：某个小程序的流程图，包括了一个执行20次的循环。假设每次循环测试时间为1ms，该小程序要完全测试所需时间 =？

- ❖ 因此，软件测试无法完全保证软件的安全性。
- ❖ 一方面是想要实现全面的测试，找出全部的错误，另一方面又要保证工程的进度，早日解决用户的问题，往往无法两全，只能在其中找到平衡点。

任何软件都是不安全的

- ❖ 关于测试，另一个问题是，全面的测试，一般情况下是针对所有可能出现的隐患进行测试，但是这需要对软件的隐患具有全方位的预见性。而在有些情况下，很多隐患是在运行期间才显露出来的，软件的开发者很难在开发阶段预见到所有可能出现的隐患，容易让测试陷入盲目。
- ❖ 因此，测试只能减少软件安全问题的发生，但是不能完全解决安全问题。
- ❖ 业界大都公认一个事实：几乎所有的软件都是带着安全隐患投入运行。

任何软件都是不安全的

- ❖ 以网络软件为例，敌方可能通过因特网获得未授权的访问的信息，或者利用软件缺陷来控制用户系统并展开攻击。
- ❖ 随着网络应用的更加丰富，用户对网络服务的依赖也相应的增加（如网上银行、网上股票、网上游戏等），这也导致了攻击的方法的增加和复杂化，从而使得安全问题更加凸显出来。
- ❖ 而软件工程师无法在开发阶段就预见到全部的攻击，提高了软件开发的难度。所谓“防不胜防”，就是这个道理。

任何软件都是不安全的

```
/*do something*/  
char *p2;  
char *p=malloc(100);  
...  
if ((p2=realloc(p,nsize))==NULL) {  
    if(p) free(p);  
    p=NULL;  
    return NULL;  
}  
p=p2;
```

思考：你对代码背后的工作真的了解吗？

任何软件都是不安全的

- ❖ 另一个解决安全问题的方法可能就是在测试前就尽量多地解决安全隐患。
- ❖ 在设计、编码阶段，熟练的软件设计人员和软件工程师完全可以尽可能多地将安全问题进行考虑并加以解决。如果在程序设计的时候就能够尽量地考虑安全问题，对软件的安全性也就会有更好的保证，可以大大减小测试的负担。

任何软件都是不安全的

- ❖ 结论：牢记任何软件都是不安全的。
- ❖ 近年来，不管是在应用方面还是在研究方面，软件安全技术越来越受到了重视，本课程将针对这些内容中的若干方面进行介绍。

软件的安全问题

- ❖ 课程内容简介
- ❖ 最新安全大数据
- ❖ 任何软件都是不安全的
- ❖ 软件不安全性的几种表现
- ❖ 软件不安全的原因
- ❖ 怎么考虑软件安全问题？

软件不安全性的几种表现

- 软件的不安全性，一般情况下的受害者就是其直接用户。
- 从用户的角度来看，软件的不安全性主要体现在两个方面。



软件不安全性的几种表现

1. 软件在运行过程中不稳定，出现异常现象、得不到正常结果、或者在特殊情况下由于一些原因造成系统崩溃。比如：

- 由于异常处理不当，软件运行期间遇到突发问题，处理异常之后无法释放资源，导致这些资源被锁定无法使用；
- 由于线程处理不当，软件运行中莫名其妙得不到正常结果；
- 由于网络连接处理不当，网络软件运行过程中，内存消耗越来越大，系统越来越慢，最后崩溃；
- 由于编程没有进行优化，程序运行消耗资源过大；等等。

见例子

软件不安全性的几种表现

某大数据处理程序需要对大规模计算结果进行分布统计，计算结果在0.00-1.00之间，估计有100万数据量，试按照0.01为分区间隔统计出各个区间的数值分布。

```
for(int i=0;i<NumberAll;i++){
    (if(a[i])<0.01)&&(if(a[i])>=0.0) R[0]++;
    (if(a[i])<0.02)&&(if(a[i])>=0.01) R[1]++;
    (if(a[i])<0.03)&&(if(a[i])>=0.02) R[2]++;
    ...
    (if(a[i])<1.00)&&(if(a[i])>=0.99) R[99]++;
}
```

思考：怎么写代码统计？

软件不安全性的几种表现

- 2. 敌方利用各种方式攻击软件，达到窃取信息、破坏系统等目的。比如：
 - 敌方通过一些手段获取数据库中的明文密码；
 - 敌方利用软件的缓冲区溢出，运行敏感的函数；
 - 敌方利用软件对数据的校验不全面，给用户发送虚假信息；
 - 敌方对用户进行拒绝服务攻击；等等。



软件不安全性的几种表现

- ❖ 通常情况下，大多数安全问题在软件运行的过程中发生，而负责软件系统运行的技术管理人员或者软件的个人用户，并不是专业的软件开发人员。
- ❖ 此时他们往往无法给出直接的应对方案，虽然可以依靠一些简单的方法，如：优化操作系统、优化网络、优化数据库管理系统或者设置额外的操作权限来对付这些剧增的安全问题，但是实际上，这些方法都是治标不治本的方法。
- ❖ 软件的生产单位需要投入大量的成本，进行软件维护。



软件的安全问题

- ❖ 课程内容简介
- ❖ 最新安全大数据
- ❖ 任何软件都是不安全的
- ❖ 软件不安全性的几种表现
- ❖ **软件不安全的原因**
- ❖ 怎么考虑软件安全问题?

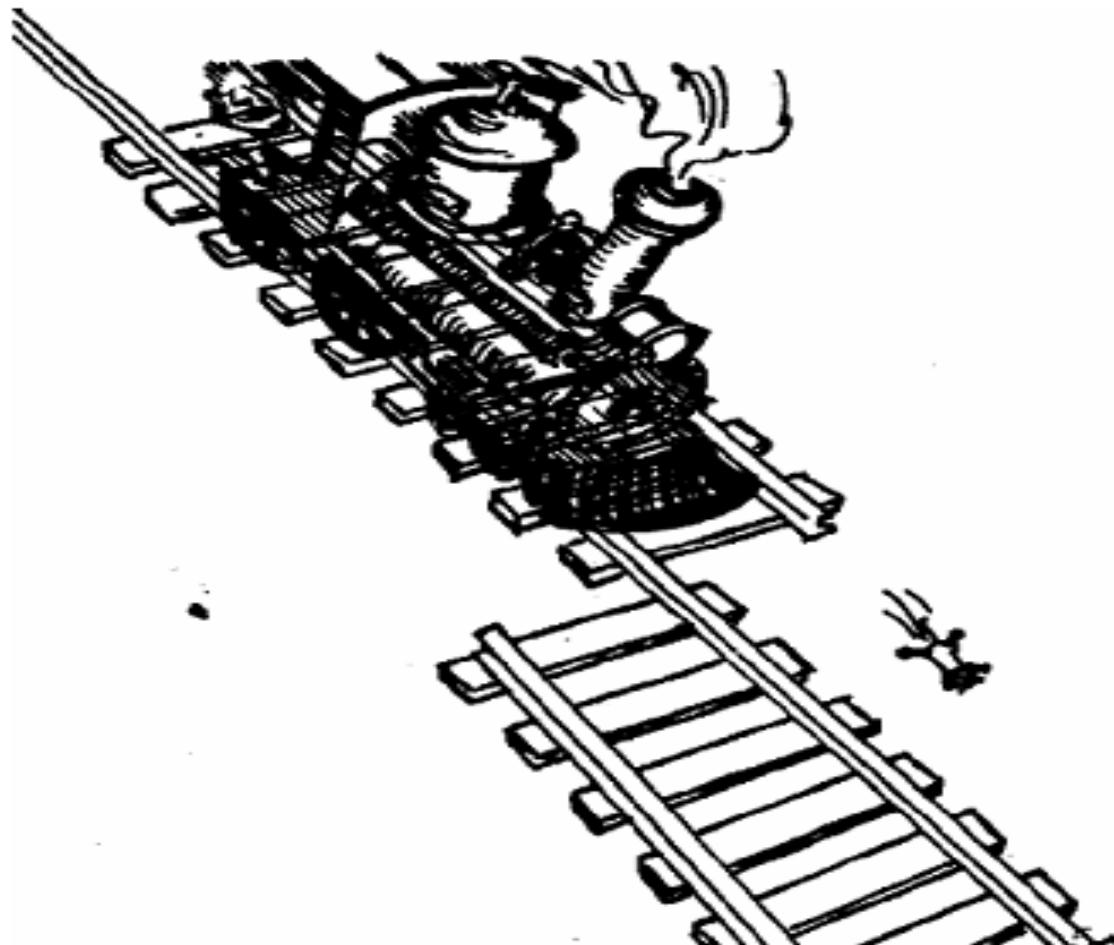
软件不安全的原因

软件出现安全问题，并造成损失，一方面是由于攻击者的猖獗，但是从开发者角度，几乎都有一个共同的基本原因：软件在设计、编码、测试和运行阶段，没有发现软件中的各种安全隐患，导致软件的不安全。

软件安全隐患一般可以分为两类： 错误和缺陷

- ❖ **错误**是指软件实现过程出现的问题，大多数的错误可以很容易发现并修复，如缓冲区溢出、死锁、不安全的系统调用、不完整的输入检测机制和不完善的数据保护措施等；
- ❖ **缺陷**是一个更深层次的问题，它往往产生于设计阶段并在代码中实例化且难于发现，如设计期间的功能划分问题等，这种问题带来的危害更大，但是不属于编程的范畴。

错误 (Error) 与缺陷 (Fault)



软件不安全的原因

- 软件不安全的原因。首先，站在软件开发者主观的角度，软件不安全的原因可以归纳为以下几种：

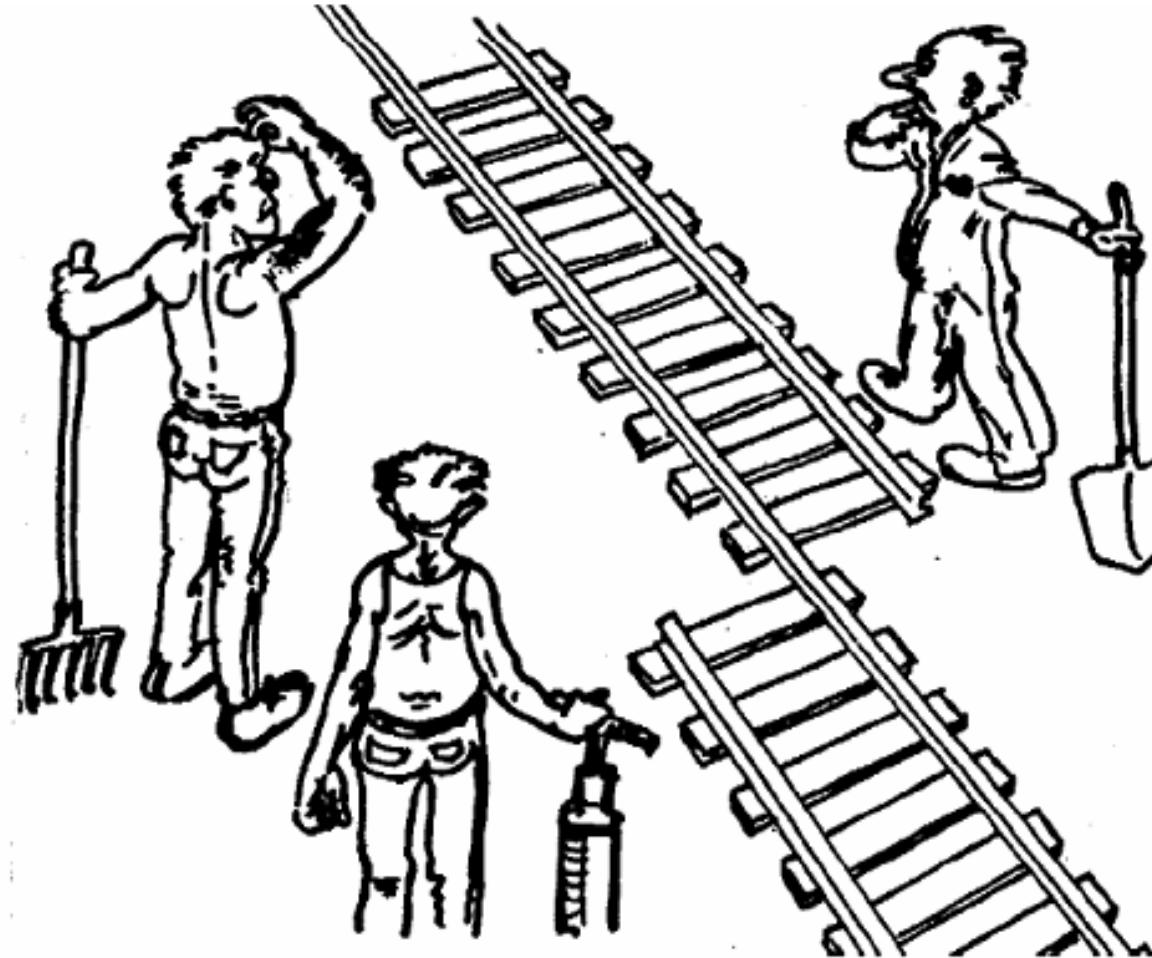
(1) **软件的生产没有严格遵守软件工程流程。**由于缺乏经验或者蓄意(如片面追求高进度)的原因，软件的设计者和开发者们没有一个统一的管理，可以在软件开发周期的任意时候，随意删除、新增或者修改软件需求规格说明书、威胁模型、设计文档、源代码、整合框架、测试用例和测试结果、安装配置说明书，使得软件的安全性保证大大减弱。

源文件1: **A(int x){ B(x) }**

...

源文件n: **B(short y){ ... }**

设计缺陷 (Design Fault)

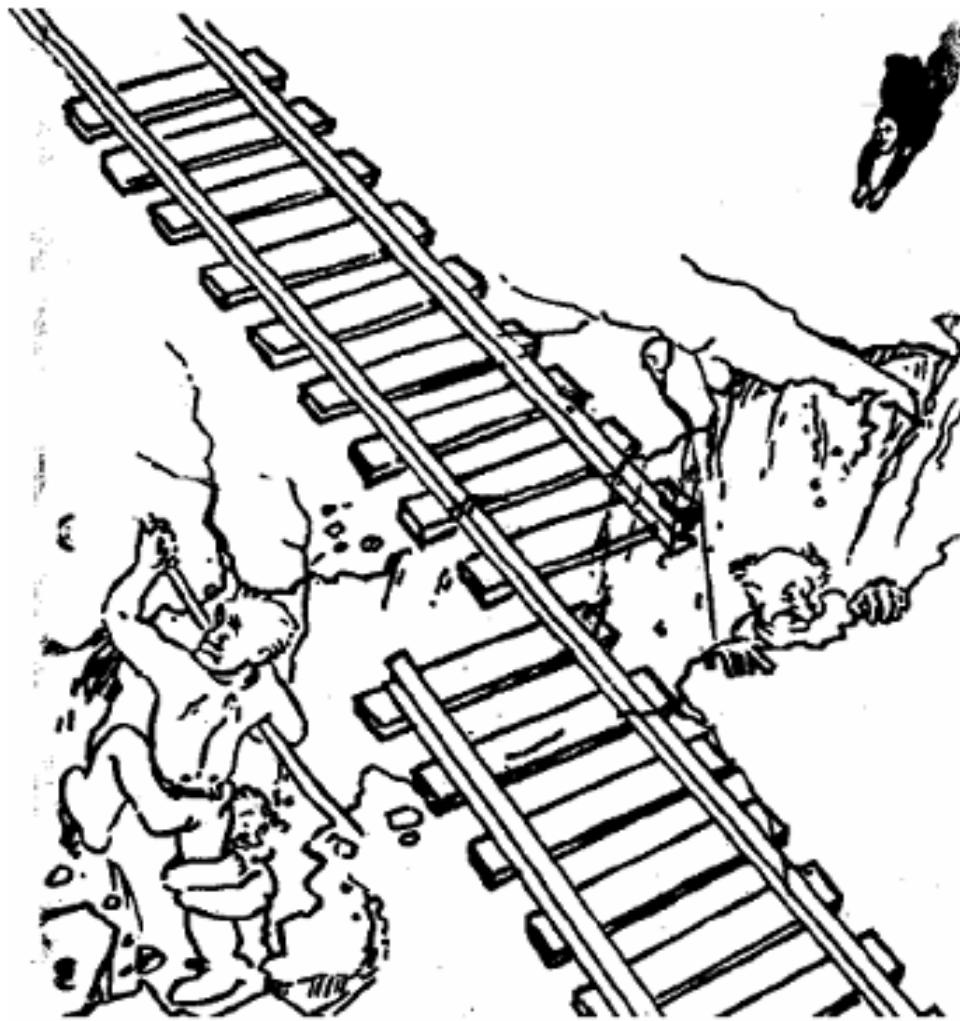


软件不安全的原因

(2) 大多数系统软件或其他商业软件，结构都相当大并且复杂，而且由于考虑到软件的扩展性，它们的设计更加巧妙，复杂性可能会更加提高一些。在运行的过程中，这些系统又可以在大量不同的状态之间转换，这个特性使得开发和使用持续正常运行的软件，是一件很困难的事情，更不用说持续安全运行了。

面对不可避免的安全威胁和风险，项目经理和软件工程师必须从开发流程做起，让安全性贯穿整个软件开发。就大多数相对成功的软件工程案例而言，如果项目经理和软件工程师针对软件缺陷进行系统的训练，可避免软件的许多安全缺陷。

复杂性导致的缺陷



软件不安全的原因

(3) 编码者没有采用科学的编码方法。在软件开发的过程中没有考虑软件可能出现的问题，仅仅将能够想到的问题停留在实验室内进行解决。实际上，有些程序，在实验室阶段根本不会出现安全隐患，如下代码：

```
int main(int argc, char* argv[])
{
    unsigned short total = strlen(argv[1]) + strlen(argv[2]) + 1;
    char* buffer = (char*)malloc(total);
    strcpy(buffer, argv[1]);
    strcat(buffer, argv[2]);
    free(buffer);
    return 0;
}
```

软件不安全的原因

(4) 测试不到位(不过有时是无法到位)。主要是测试用例的设计无法涵盖尽可能典型的安全问题。如下的登录表单：

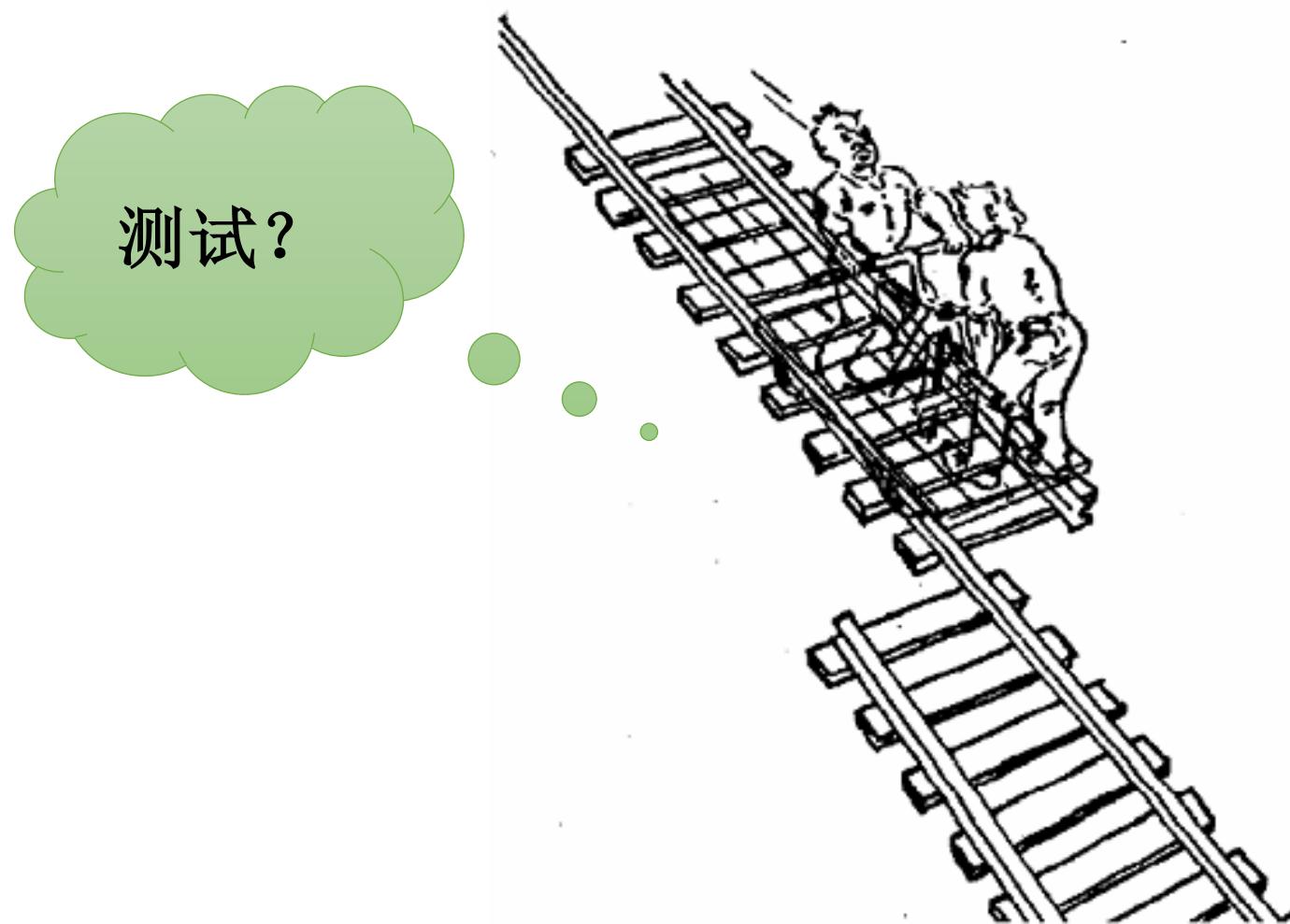
用户名

密码



一般测试用例只是设计输入正确的用户名和密码，看能否正常登录；再输入错误的用户名和密码，看能否得到相应的错误提示。但是攻击者如果输入某些和**SQL注入**有关的值，就有可能在不需要知道用户名和密码的情况下登录到系统，甚至知道系统中的其他信息或对系统中的内容进行修改。

测试不到位



软件不安全的原因

从软件工程客观角度讲，软件的安全性隐患又来源于以下几个方面：

(1) 软件复杂性和工程进度的平衡。如前所述，软件规模复杂了，不仅仅是编码工作量的提高，更重要的是其中需要考虑的问题更加复杂，测试用例规模也呈指数级增长。但是工程进度只是按照软件规模进行适当的延长，因此很多问题来不及解决，软件带着缺陷投入使用。

软件不安全的原因

- (2) 安全问题的不可预见性。主要是软件工程师对运行的实际情況的不了解，在测试时作出过于简单的假设。有些問題，包括对軟件的功能、输出和软件运行环境的行为状态，或者外部实体(用户，软件进程)的预期输入，都无法完全考慮到。而攻击者有足够的時間进行攻击方法的研究。
- (3) 由于软件需求的变动。软件規格说明书或设计文档无法一开始就確定下来；在现代软件工程中，很多软件的需求变动，导致其设计本来就是变动的，很多安全問題可能在变动的过程中被忽略。

软件不安全的原因

(4) 软件组件之间的交互的不可预见性。如客户可能在运行软件的过程中，自行安装第三方提供的组件，开发者根本无法知道客户的软件将要和谁交互，软件在运行的过程中出现安全问题。



软件不安全的原因

- ❖ 因此，我们可以看到，不管采用了什么样的措施，软件的安全问题都无法完全避免。
- ❖ 即使在需求分析和设计时可以避免（如通过形式化方法），或者在开发时可以避免（比如通过全面的代码审查和大量的测试），但缺陷还是会在软件汇编、集成、部署和运行时候被引入。
- ❖ 不管如何忠实的遵守一个基于安全的开发过程，只要软件的规模和复杂性继续增长，一些可被挖掘出来的错误和其他的缺陷是肯定存在的。我们所能做的工作就是尽量让安全问题变少，而不能完全消灭安全问题。

软件的安全问题

- ❖ 课程内容简介
- ❖ 最新安全大数据
- ❖ 任何软件都是不安全的
- ❖ 软件不安全性的几种表现
- ❖ 软件不安全的原因
- ❖ 怎么考虑软件安全问题?

软件安全防护手段

- 隔离
- 内存安全语言
- 形式化验证
- AI for Software Security

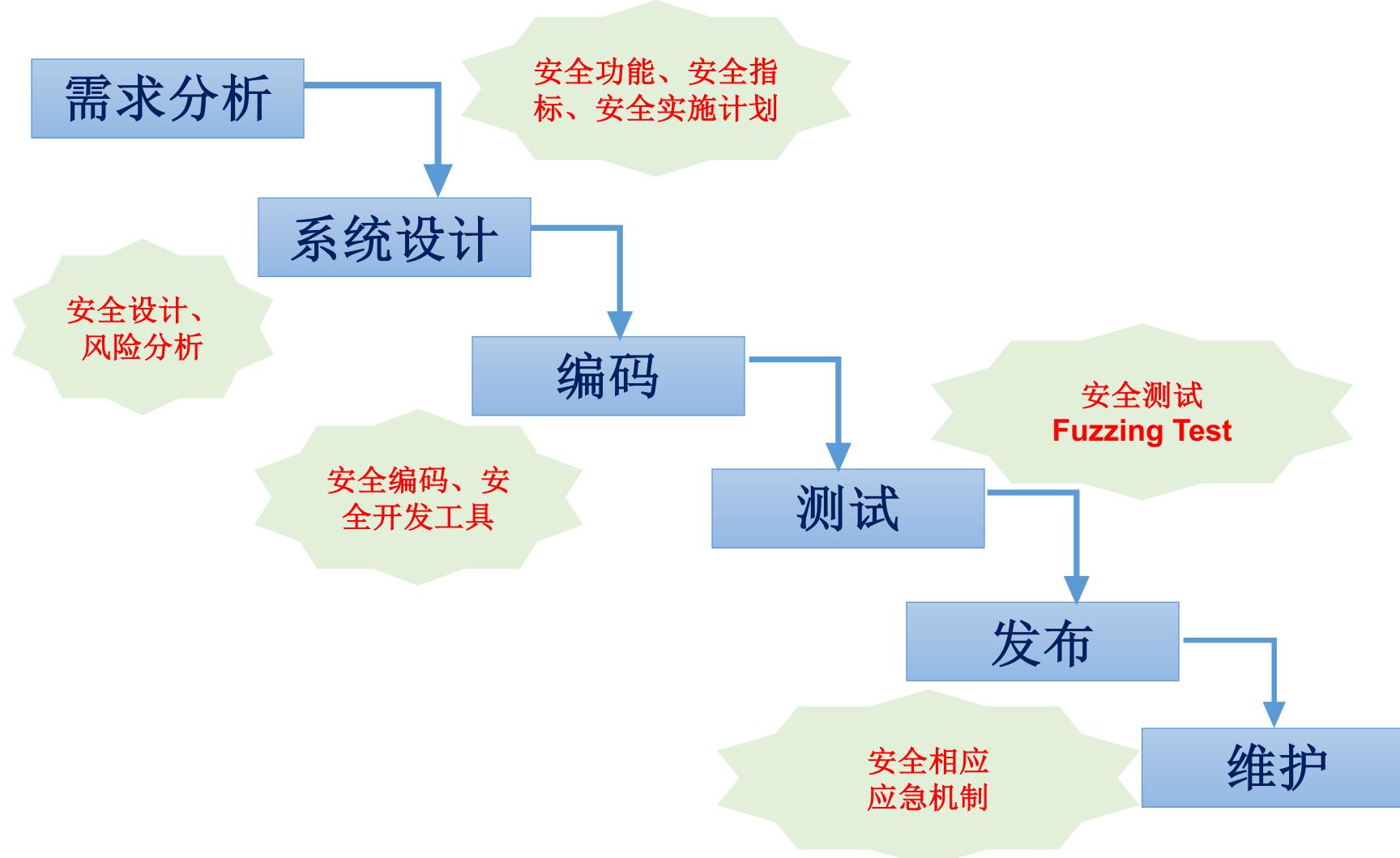
软件安全防护手段

- 安全设计与开发
- 保障运行环境
- 加强软件自身行为认证
- 恶意软件检测与查杀
- 黑客攻击防护
- 系统还原
- 虚拟隔离等。

1. 安全设计与开发

- 强化软件工程思想，将安全问题融入到软件的开发管理流程之中，在软件开发阶段尽量减少软件缺陷和漏洞的数量。
- 微软：信息技术安全开发生命周期流程（Secure Development Lifecycle for Information Technology，缩写为SDL-IT）。
 - 该流程包含有一系列的最佳实践和工具，用于微软内部业务应用以及许多微软客户的开发项目中。
 - 微软的Windows 7、8、10系统
- 华为SDL实践
 - 需求、设计、开发阶段，上线前测试时，应急响应，供应链安全

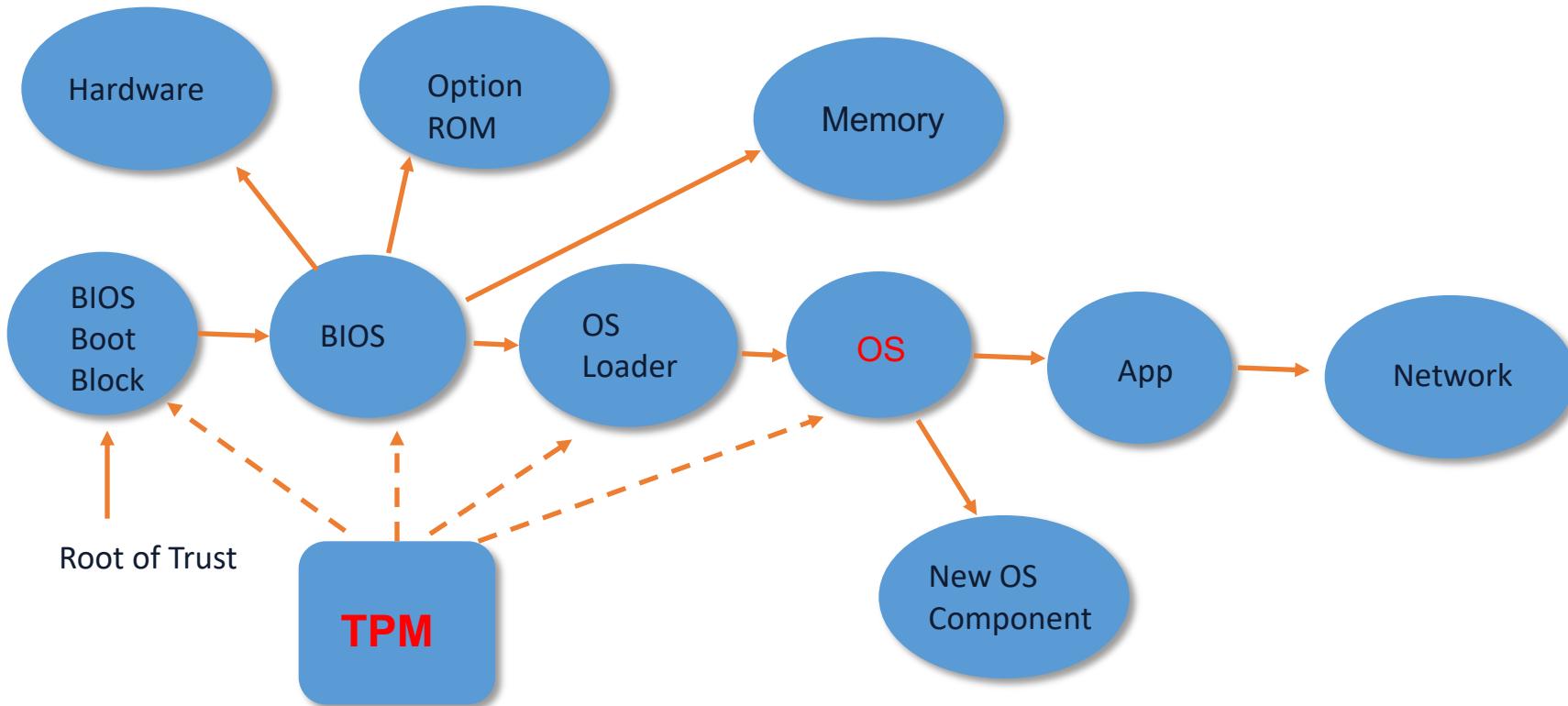
SDL开发模式



2. 保障运行环境

- 保障软件自身运行环境，加强系统自身的数据完整性校验
 - 软件完整性校验
 - 目前很多安全软件在安装之初将对系统的重要文件进行完整性校验并保存其校验值，如卡巴斯基安全套件。
 - 系统完整性校验
 - 目前有些硬件系统从底层开始保障系统的完整性，可信计算思想是典型代表。

TCG的可信计算信任链的传递

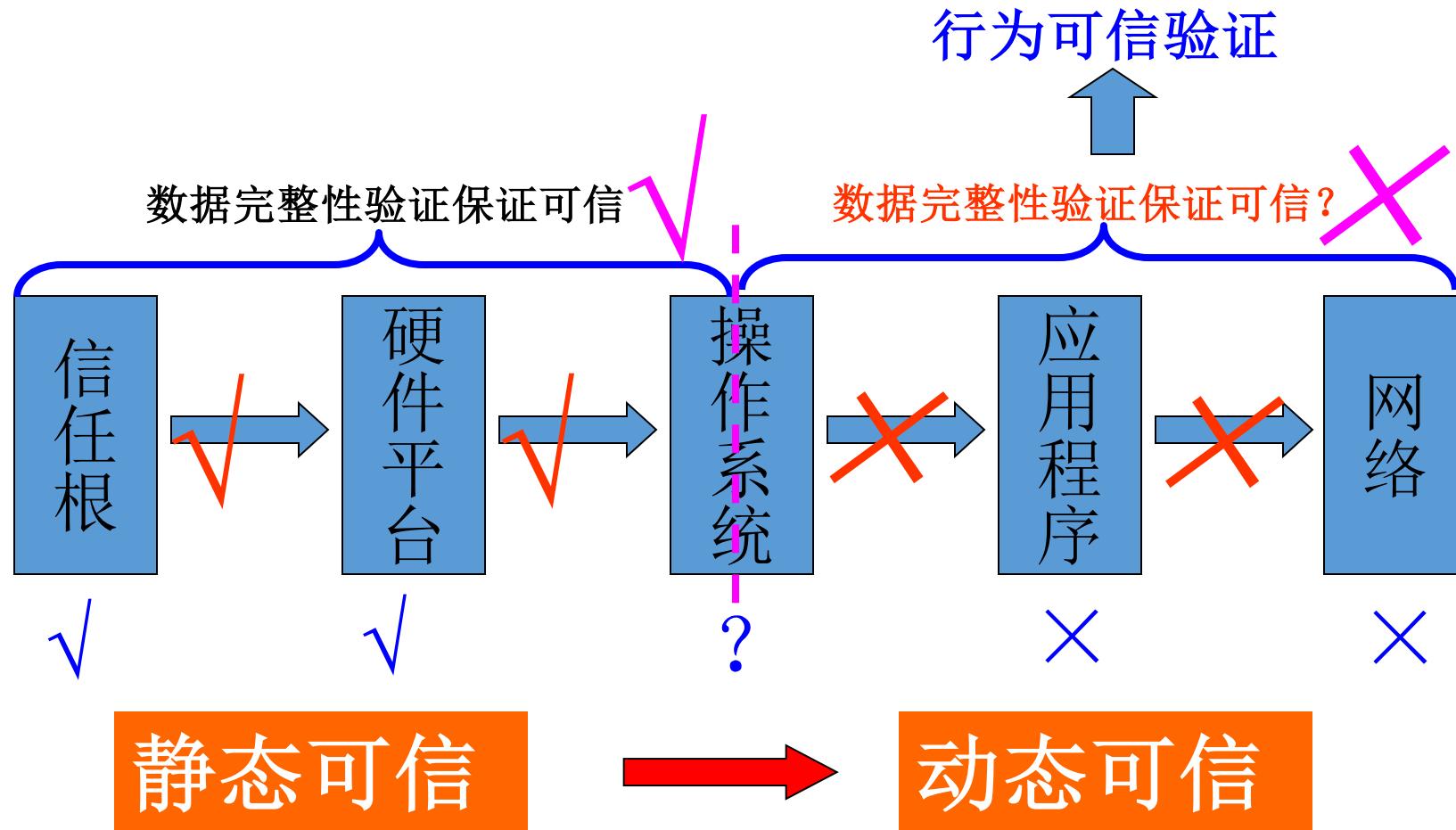


3. 加强软件自身行为认证

□ 软件动态可信认证

- 在确保软件数据完整性的前提下，如何确保软件的行为总是以预期的方式，朝着预期的目标运行。

信任链的传递



高可信软件技术研究

- 美国计算研究协会: 把高可信软件系统看作是目前计算机研究领域必须应对的五大挑战之一。
- 美国国家科技委员会: 在其总统财政预算报告中指出, 高可信软件技术是需要优先开展的研究工作, 包括构造更加安全、可靠和健壮的可信软硬件平台, 提供更高效的可信软件开发技术, 以及建立新的保证复杂软件系统高可信的科学和工程体系等。
- 美国国防部高级研究计划署 ([Defense Advanced Research Projects Agency, DARPA](#)) : 将高可信系统和软件列为中国需要面对的四大挑战之一。
- 美国国家科学基金会、美国宇航局和美国安全局 ([National Security Agency, NSA](#)) 等: 高可信软件技术研究的重要投资方。
- 微软: 可信赖计算([Trustworthy computing, TWC](#))

可信软件

- 我国政府十分重视软件系统的可信性问题。
 - 国家自然科学基金委从2007年启动了“可信软件基础研究”重大研究计划；
 - 国家高技术发展（863）计划中设立了专门的重大项目，研究高可信软件生产工具及集成环境；
 - 国家重点基础研究发展（973）计划将可信软件的研究确定为重点发展方向，研究基于网络的复杂软件可信度和服务质量。

4. 恶意软件检测与查杀

- 反病毒软件主要用来对外来的恶意软件进行检测。
 - 通常采用病毒特征值检测、虚拟机、启发式扫描、主动防御、云查杀等等几种方法来对病毒进行检测。

- 恶意软件是软件安全的一个主要安全威胁来源，针对系统的外来入侵通常都离不开外来恶意软件的支撑。

5. 黑客攻击防护

- 防火墙
 - 网络、主机防火墙
- 入侵检测系统IDS
- 入侵防护系统IPS
 - 基于网络、基于主机（HIPS）
- 基于主机的漏洞攻击阻断技术
 - EMET: [Microsoft's Enhanced Mitigation Experience Toolkit](#)

6. 系统还原

- 将关键系统文件或指定磁盘分区还原为之前的备份状态，从而将已有系统中的恶意程序全部清除，以保护系统安全。
 - Windows自带的“系统还原”功能
 - Ghost还原软件
 - 还原卡、影子系统（PowerShadow）等

7. 虚拟隔离等

□虚拟机（如VMware）

■隔离风险

□用户可以通过在不同的虚拟机中分别进行相关活动（如上网浏览、游戏或网银等重要系统登陆），从而可以将危险行为隔离在不同的系统范围之内，保障敏感行为操作的安全性。

□沙箱，也叫沙盘或沙盒（如SandBox）

■隔离风险

■通常用于运行一些疑似危险样本，从而可以隔离安全威胁，也可以用于恶意软件分析。

上述是目前主要的措施，还有很多...

- 软件行为审计
- 冗余软件机制
- 拟态软件
- ...

课后思考

- Safety与Security的区别是什么？
- 软件安全问题为何日益严重？为什么说软件一定是不安全的？
- 软件安全防护手段有哪些？它们各自从哪些角度来保障信息安全？

本章小结

- ❖ 课程内容简介
- ❖ 最新安全大数据
- ❖ 任何软件都是不安全的
- ❖ 软件不安全性的几种表现
- ❖ 软件不安全的原因
- ❖ 怎么考虑软件安全问题？