

# 第1讲. 软件安全概述

网络空间安全学院 付才

**Mail:** [fucai@hust.edu.cn](mailto:fucai@hust.edu.cn)

**QQ: 5146279**



# 最新软件安全态势

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

# 2021最新软件安全态势

## (1) 2021.12, log4j史诗级巨大漏洞

"史上最大漏洞",被中国程序员发现 Log\_阿帕奇\_迈卡菲



2021年12月15日 美联社则评论称,这一漏洞可能是近年来发现的最严重的计算机漏洞。Log4j在全行业和政府使用的云服务器和企业软件中“无处不在”。除非被修复,否则犯罪分子、间谍乃至编程信守,都可以...

搜狐网 百度快照

Log4j漏洞最早由阿里云团队发现;HashiCorp上市,市值1...

2021年12月15日 Log4j漏洞最早由阿里云团队发现 HashiCorp 挂牌上市,市值 152 亿美元,成为今年全球市值最高的开源公司 从Mac App Store 下载的 Xcode 13.2 存在 Bug GitLab ...

网易新闻 百度快照

Java的日志记录工具log4j发现(史诗级漏洞) 悟空大师的博客



2021年12月10日 Java的日志记录工具log4j发现史诗级漏洞个Apache Log4j2反序列化远程代码执行漏洞细节已被公在JNDI注入漏洞,当程序将用户输入的数据进行日志记录

CSDN技术社区 百度快照

**阿里云发现Log4j2后未及时报告 被工信部网络安全威胁信息共享平台暂停合作**

2021-12-22 11:11

前言: 阿里云公司发现阿帕奇 (Apache) Log4j2组件严重安全漏洞隐患后, 未及时向电信主管部门报告, 未有效支撑工信部开展网络安全威胁和漏洞管理, 现暂停阿里云合作单位6个月。

漏洞依旧不断发现, 但世界变化很快

log4j史诗级巨大漏洞? 原来是这样<https://zhuanlan.zhihu.com/p/447266098>

# 2021最新软件安全态势

## (2) 2021.7, Kaseya 事件与供应链风险

2021 年 7 月初，IT 管理软件供应商 Kaseya 发生的安全事件，再次凸显了企业面临来自 IT 供应链中供应商的威胁正日益加剧。该事件后来归因于 REvil/Sodinokibi 勒索软件组织的一个附属机构，其中涉及威胁行为者利用 Kaseya 虚拟系统管理员（VSA）技术中的三个漏洞，而许多托管服务提供商（MSP）使用该技术来管理其客户的网络。攻击者利用这些漏洞，使用 Kaseya VSA 在属于 MSP 下游客户的数千个系统上分发勒索软件。

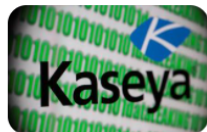
### 美国软件商Kaseya遭REvil勒索软件供应链攻击



2021年7月5日 美国东部时间周五下午 2 点左右Kaseya 被攻击，2021 年 7 月 3 日晚上 7:30 和晚上 9:00 更新、7 月 4 日上午 10:00 Kaseya 连发三次警告.....2021 年 7 月 4 日美国东部时间上午 10:...

祺印说信安 百度快照

### 在遭到勒索软件攻击前几年,Kaseya 就已经收到了安全警告



2021年7月11日 IT之家 7 月 11 日消息 远程 IT 服务管理软件 Kaseya VS A 近日遭到了勒索软件攻击，被入侵了 100 多万台电脑，攻击者索要 70 00 万美元。据彭博社，Kaseya 的前员工称，他们在 2017 ...

IT之家 百度快照

# 2021最新软件安全态势

## (3) 2021.5, Colonial Pipeline 攻击&国家安全

2021 年 5 月，针对美国**基础设施**管道运营商 Colonial Pipeline 的**勒索软件**攻击占据了新闻头条，此举对美国广大民众造成了广泛影响：中断了数百万加仑燃料的运输，并引发了美国东海岸大部分地区的短暂性天然气短缺。这起事件也成功将勒索软件提升为国家安全级别的问题，并引起了白宫的关注。事件发生几天后，拜登总统发布了一项行政命令，要求联邦机构实施新的控制措施以加强网络安全。。

[美国因Colonial Pipeline遭勒索攻击而宣布紧急状态,网络“...](#)



2021年5月13日 上周,美国最大燃油管道运营商Colonial Pipeline被网络攻击致使其被迫关闭所有输油管道一时间成为国内外焦点,引发了国内外对于国家关键基础设施的高度重视。 图源:Wired 美国最大燃...

知乎 百度快照

[Colonial Pipeline攻击事件初步调查结果\(未及时修复Exchange...\)](#)



2021年5月13日 Nicole Perlroth 在推文中指出,关于 Colonial Pipeline 的取证发现,他们仍在使用存在漏洞的微软 Exchange 版本,以及其他明显的缺陷。网络安全和基础设施安全局警告管道运营商, 20...

cnBeta 百度快照

# 2021最新软件安全态势

## (4) 2021.12, 开源代码**投毒**&攻防新态势

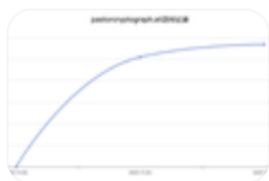
12月23日 03:32 攻击者在NPM官方仓库上传了radar-cms 恶意包(https://www.npmjs.com/package/radar-cms), 包代码和主页完全复制TypeChain正常包, 有可能是为了绕过信誉检查, 目前已下载40余次。radar-cms包 恶意功能触发方式相对常见, 在package.json 中的postinstall字段添加了一段恶意命令, 功能是在安装radar-cms包时, 窃取 kubeconfig文件、kerberos凭据、/etc/passwd、/etc/hosts等敏感文件。

### 比Log4j漏洞还可怕? NPM代码库被开发者“投毒”

2022年1月11日 近日, 流行软件包管理工具NPM的两个开源库“color”和“faker”被其开发者“投毒”。这些库的开发者故意引入了一个无限循环, 破坏了数千个依赖“color”和“fak...”

GoUpSec 百度快照

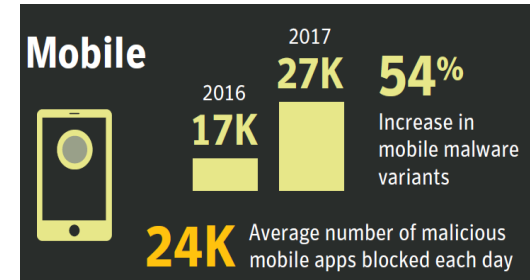
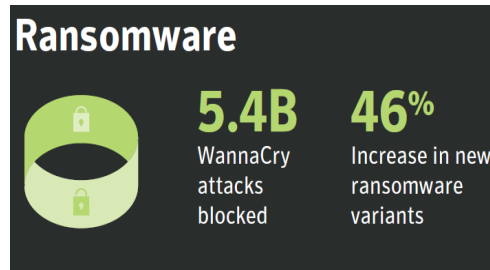
### 安全通知|NPM官方仓库遭遇coa等恶意包投毒攻击\_Tencent...



2021年11月5日 今天,腾讯洋葱入侵检测系统发现开源软件沙箱主机出现异常行为,跟进发现npm官方仓库的coa、rc被投毒攻击已经在腾讯软件源紧急拦截相关版本。目前npm官方仓库已将相应版本的恶意包删...

CSDN 技术社区 百度快照

# Symantec (Latest) Internet Security Threat Report



2018



# Symantec (Latest) Internet Security Threat Report

## Supply chain attacks



2018



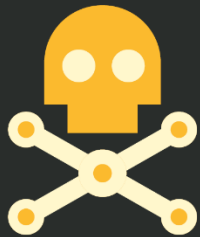
## SUPPLY CHAIN ATTACKS

78%↑



# Symantec (Latest) Internet Security Threat Report

## Malware



**92%**

Increase  
in new  
downloader  
variants

**80%**

Increase  
in new  
malware  
on macs

**8,500%**

Increase in  
coinminer  
detections

2018



## MALICIOUS EMAIL

**48%**

OF MALICIOUS EMAIL ATTACHMENTS  
ARE OFFICE FILES, UP FROM 5% IN 2017

## POWERSHELL

**1000%**

INCREASE IN  
MALICIOUS  
POWERSHELL  
SCRIPTS

# 恶意代码发展态势

## 安全威胁大数据:



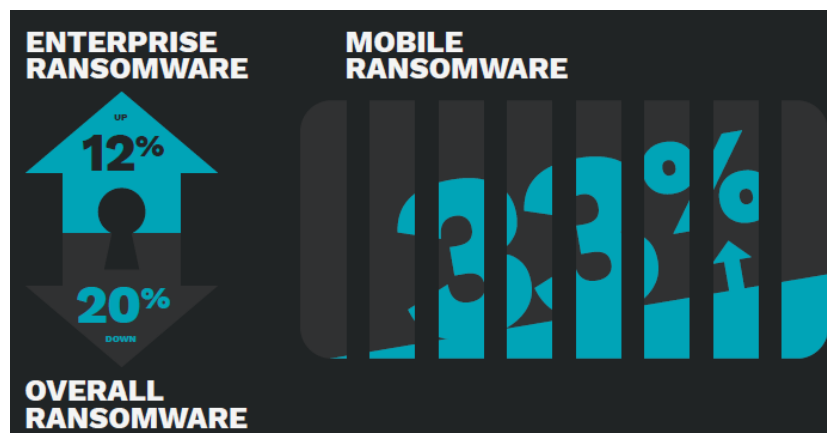
## Malware

**92%**  
Increase in new downloader variants

**80%**  
Increase in new malware on Macs

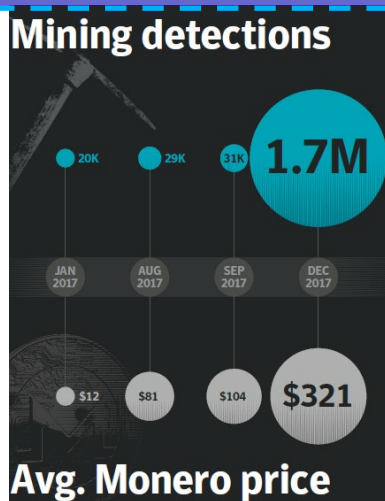


2018

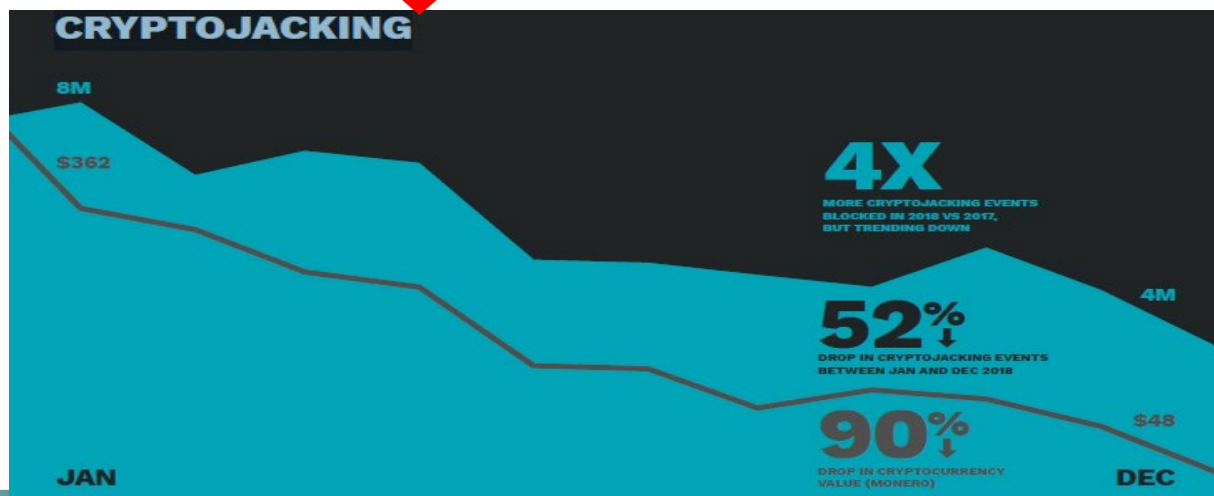


# 恶意代码发展态势

2018

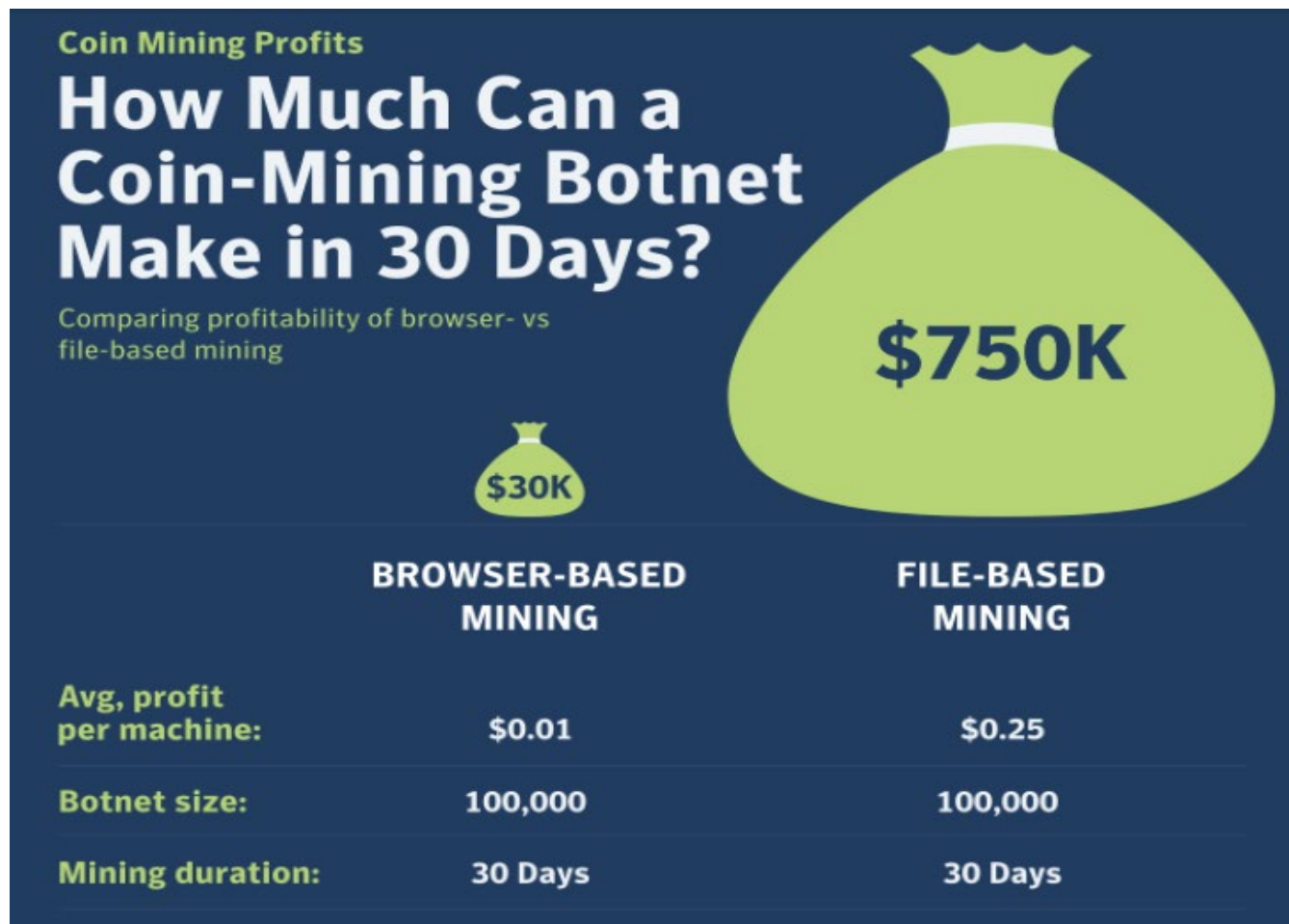


**Cryptojacking:  
A Modern  
Cash Cow**

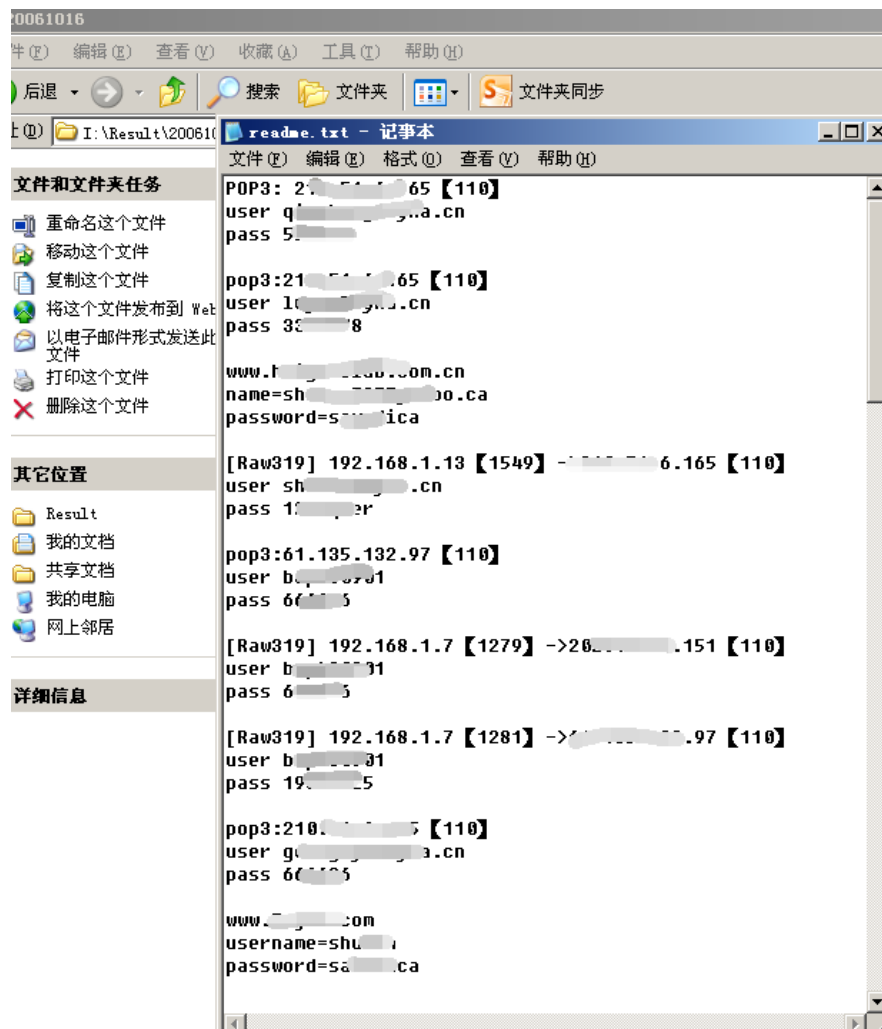


挖币恶意代码，新时代！

# 恶意代码发展态势



# 恶意代码发展态势



关于我自己

# 软件的安全问题

- ❖ 软件安全道路依旧任重道远
- ❖ **进入到我们的《软件安全》课程...**

# 软件的安全问题

- ❖ **课程内容简介**
- ❖ 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？

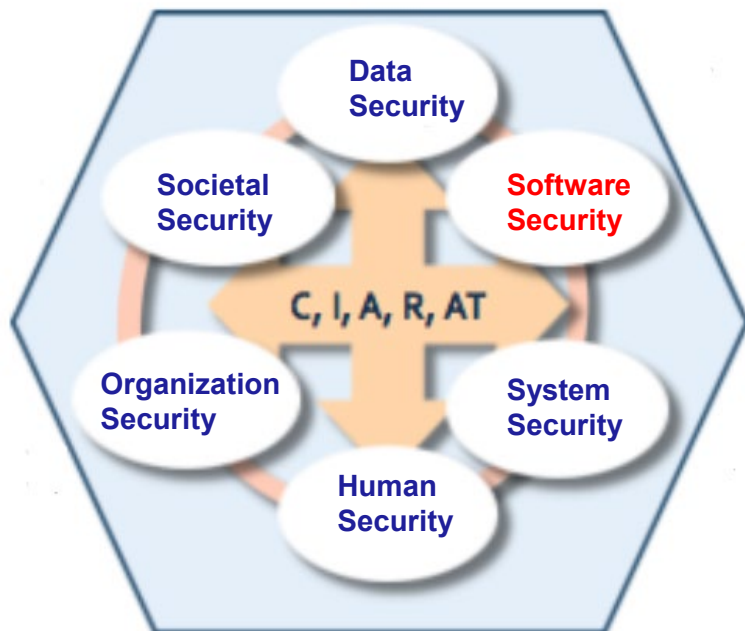
# 简介

- ❖ 软件，是组成计算机应用的重要组成部分，当软件由于不安全而遭受攻击，或者运行期间出现错误时，会给用户带来巨大的损失。如犯罪分子利用软件漏洞来获取有价值的信息，用于牟取利益；又如软件因为开发时没有考虑运行时的具体情况，而造成运行的突然崩溃；等等。
- ❖ 面对越来越频繁的软件安全隐患带来的损失，对软件的开发——软件工程师，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，切实为软件使用者服务。
- ❖ **漏洞如何产生？怎么分析与利用？又怎么防御？**
- ❖ **软件中的“异类” - 恶意代码现在越来越影响到人们的日常生活，它们如何工作？如何防御？如何分析？**
- ❖ 本课程主要就是针对这些问题进行介绍，了解软件安全的核心技术，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。





# 简介



- ◆ **CSEC2017**模型有六个知识领域：数据安全，**软件安全**，系统安全，人员安全，组织安全以及社会安全
- ◆ **CSEC2017**思想模型包括学科五个内涵属性：**C** -机密性，**I**-完整性,**A**-可用性,**R**-风险,**AT**-对抗

软件安全是指采取工程的方法使得软件在敌对攻击的情况下仍能继续正常工作，即采用系统化、规范化和可量化的方法来构建安全的软件并实施安全防护。



# 简介

## ➤ 第一部分 基础（1-2周）

- ❖ **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** 系统引导与控制权、X86处理器的工作模式、OS内存结构、磁盘结构、FAT32文件系统以及软件逆向与工具

## ➤ 第二部分 恶意代码（3-7周）

- ❖ **3. 恶意代码及其分类**
- ❖ **4.PE与Windows PE病毒**
- ❖ **5.宏病毒与脚本病毒**



# 简介

❖ 6.网络蠕虫

❖ 7.木马与后门

❖ 8.恶意代码检测与防范

❖ ➤ 第三部分 软件脆弱性（7-12周）

❖ 9.软件漏洞概述

❖ 10. 漏洞分析 栈/堆/整数/格式化溢出，PWN，软/硬件防范等

❖ 11.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



# 考试安排

- **综合评分**

- 考试70%(闭卷)
- 平时30%(课堂15%+作业15%)
- 实验课为独立课程

- **闭卷，强调实践**

- 上课讲授的一定要是自己动手实践
- 考试即便可以百度也没有答案



# 课程要求

- 遵守课堂纪律
  - 按时到课
  - 关闭手机(在线课堂没有带电脑除外)
  - 保持安静(回答问题除外)
- 认真完成实验
  - 验证类试验
  - 综合类试验
  - 开发类试验
  - 提交实验报告
- 请学委建立QQ群, 交流、活跃课程气氛(已有)

# 主要参考资料

- **主教材:**

**软件安全, 超星课堂电子教案, 计划正式出版时间2022.12**

**参考资料:**

- **软件安全 彭国军, 武汉大学出版社**
- **计算机病毒原理与防治技术, 韩兰胜, 刘铭, 彭冰、付才, 华中科技大学出版社**
- **郭克华、王伟平, 软件安全实现——安全编程技术. 北京: 清华大学出版社, 2010**
- **Hacking: The Art of Exploitation, 2nd Edition, Jon Erickson, No Starch Press, January 15, 2008, 美国**
- **Oday安全: 软件漏洞分析技术, 王清, 电子工业出版社 2008.4.1**

# 软件的安全问题

- ❖ 课程内容简介
- ❖ **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？

# 思考：网络空间到底是什么空间？

## ● 软件定义万物

- ◆ 1968年，绕月的阿波罗8号飞船升空第5天，宇航员误操作删除了所有导航数据，致使飞船无法返航。程序员们连夜奋战9小时，设计出了一份新导航数据并经由巨大的地面天线阵列上传到阿波罗8号，让它顺利返航。
- ◆ 现在：OTA “Over-The-Air”

## ● 网络空间两个子空间：

⌘ 代码子空间

⌘ 数据子空间

## ● 元宇宙

⌘ 元宇宙是什么空间？



软件定义

软件定义存储

软件定义网络

软件定义汽车

软件定义边界SDP

软件定义无线电

软件定义一切

软件定义硬件

件与硬件分隔开的  
域网络(SAN)系统  
消除了软件对...

什么是软件定义



[www.redhat.com/zh/topics/data-...](http://www.redhat.com/zh/topics/data-...) 百度快照

GoldVIP加速了软件定义汽车的芯片评估和软件开发



使客户能在几分钟内评估S32G芯片的价值,对硬件进行抽象,使开发人员能专注于软件创新,简化开发并实现新的汽车应用程序的快速原型制作

恩智浦(中国)管理 2022-03 广告 百度快照

何谓“软件定义” - 知乎

2020年6月18日 所谓软件定义,就是用软件去定义硬件的功能,用软件为硬件赋能。软件定义的核心是API(Application Programming Interface)。在API之上,一切皆可编程;API之下,“如...

知乎 百度快照



# 软件安全成为网络空间核心支持

- 当元宇宙与物理世界高度融合，世界将越来越网络空间化，这个由代码与数据构成的空间，意味着软件安全越来越重要。

# 安全（Safety vs Security）

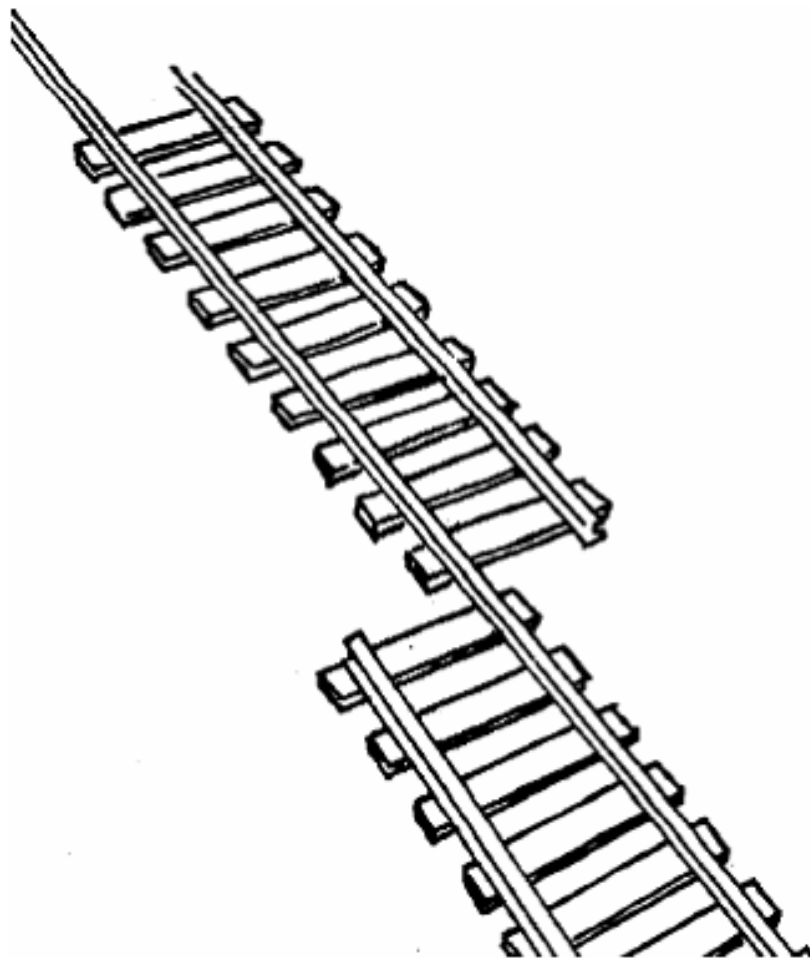
- **Safety**

- 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 param,
                    unsigned int w, unsigned int h)
{
    unsigned int temp;    unsigned int Area;
    g_exam = para;
    Area=RECT2(w+param,h);
    temp = square_exam ( Area,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

## ● 先看看怎么知道发生了软件安全问题：

- 使用某些交易软件的过程中，某些敏感信息，如个人身份信息、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，软件的开发具备两个新的挑战：
  - ❖ 软件复杂性加强
  - ❖ 可扩展性要求的提高

# 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。

# 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？



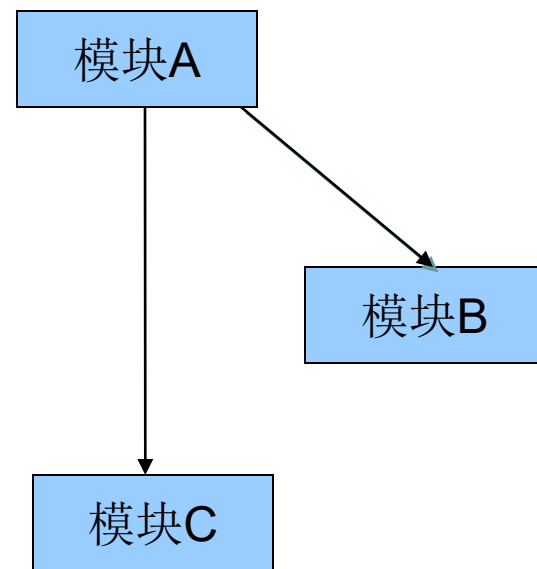
# 任何软件都是不安全的

## ● 一般怎样解决这些安全问题？

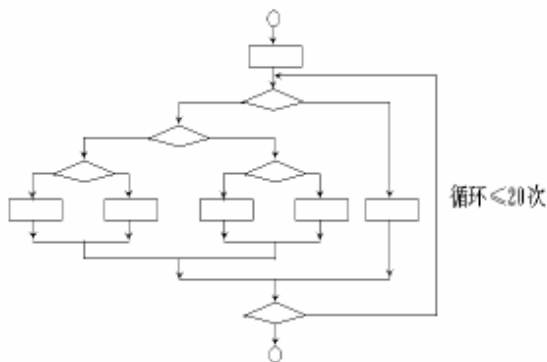
- ❖ 大多数人首先可以想到的方法是软件测试，通过测试来减少软件中的缺陷。
- ❖ 但是，由于软件系统规模越来越大，软件开发的进度要求越来越高，不可能在有限的时间内考虑所有安全方面的问题，即使进行了全方位的测试，也只能对所有的测试案例进行很小范围的覆盖。

# 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 $X*Y*Z$ 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。假设每条路径测试时间为**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

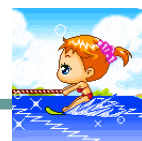
- ❖关于测试，另一个问题是，全面的测试，一般情况下是针对所有可能出现的隐患进行测试，但是这需要对软件的隐患具有全方位的预见性。而在有些情况下，**很多隐患是在运行期间才显露出来的**，软件的开发很难在开发阶段预见到所有可能出现的隐患，容易让测试陷入盲目。
- ❖因此，测试只能减少软件安全问题的发生，但是不能完全解决安全问题。
- ❖业界大都公认一个事实：**几乎所有的软件都是带着安全隐患投入运行。**

# 任何软件都是不安全的

- ❖ 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

- ❖ 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,nsiz)) == 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

- ❖ 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
- ❖ 任何软件都是不安全的
- ❖ **软件不安全有哪些表现?**
- ❖ 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。
- ❖ **软件的生产单位就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
- ❖ 任何软件都是不安全的
- ❖ 软件不安全有哪些表现?
- ❖ **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。**

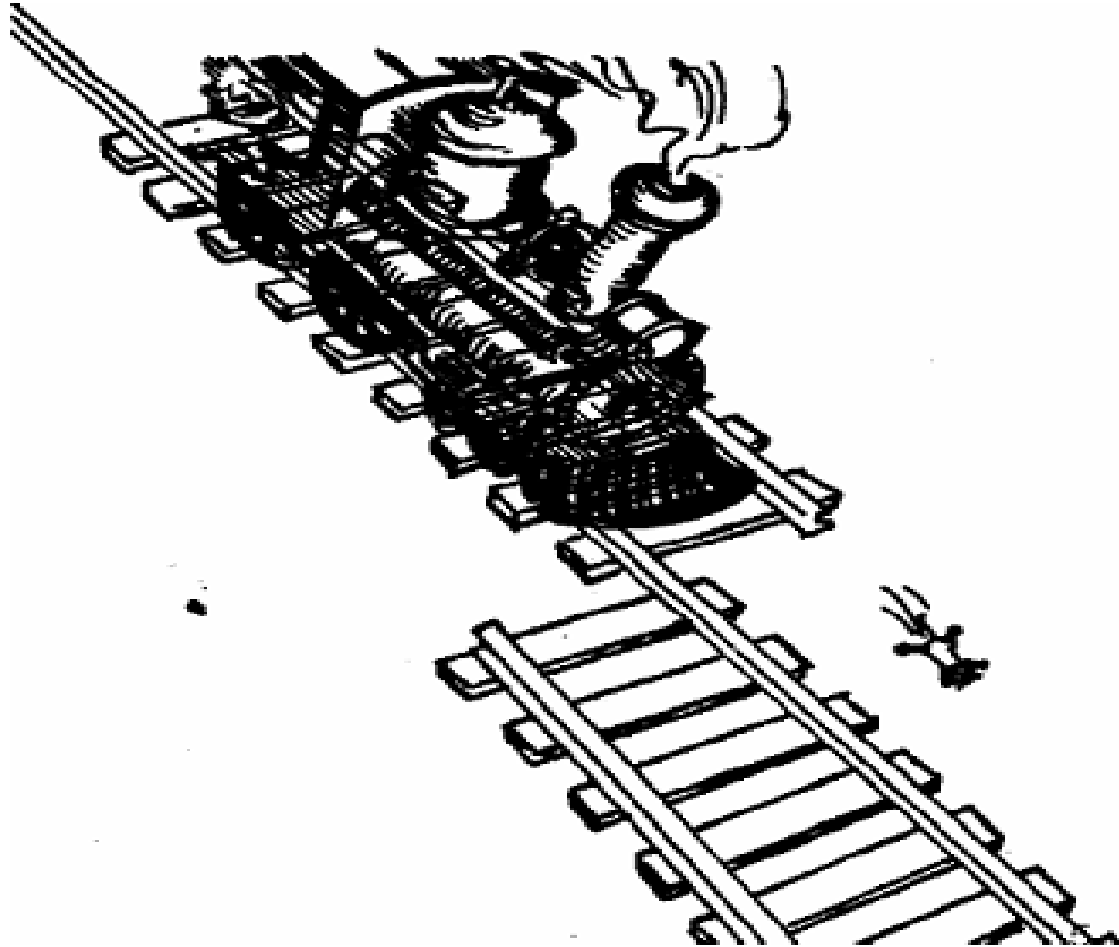
❖ **软件安全隐患一般可以分为两类： 错误和缺陷**

❖ **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，站在软件的开发者主观的角度，软件的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)



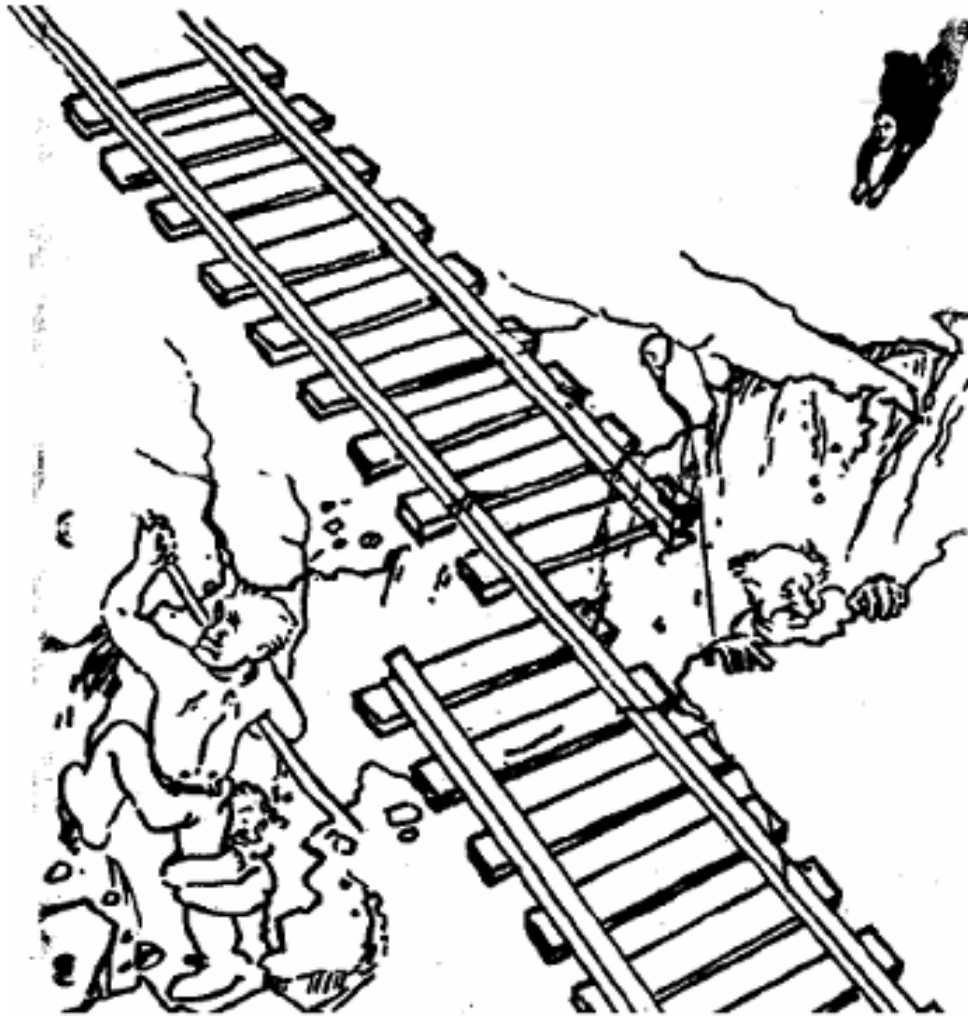
# 软件不安全的原因

(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的始终。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，可以避免软件的许多安全缺陷。



# 复杂性导致的缺陷



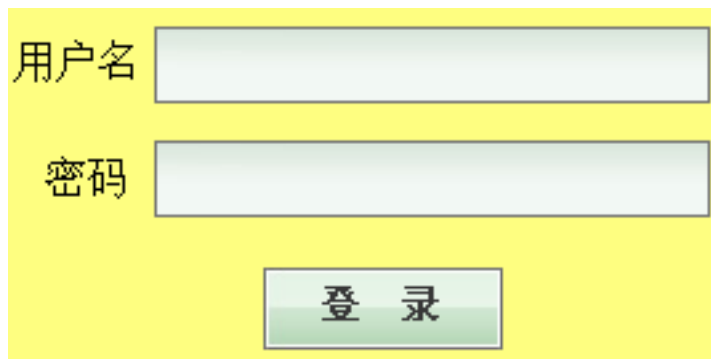
# 软件不安全的原因

**(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：



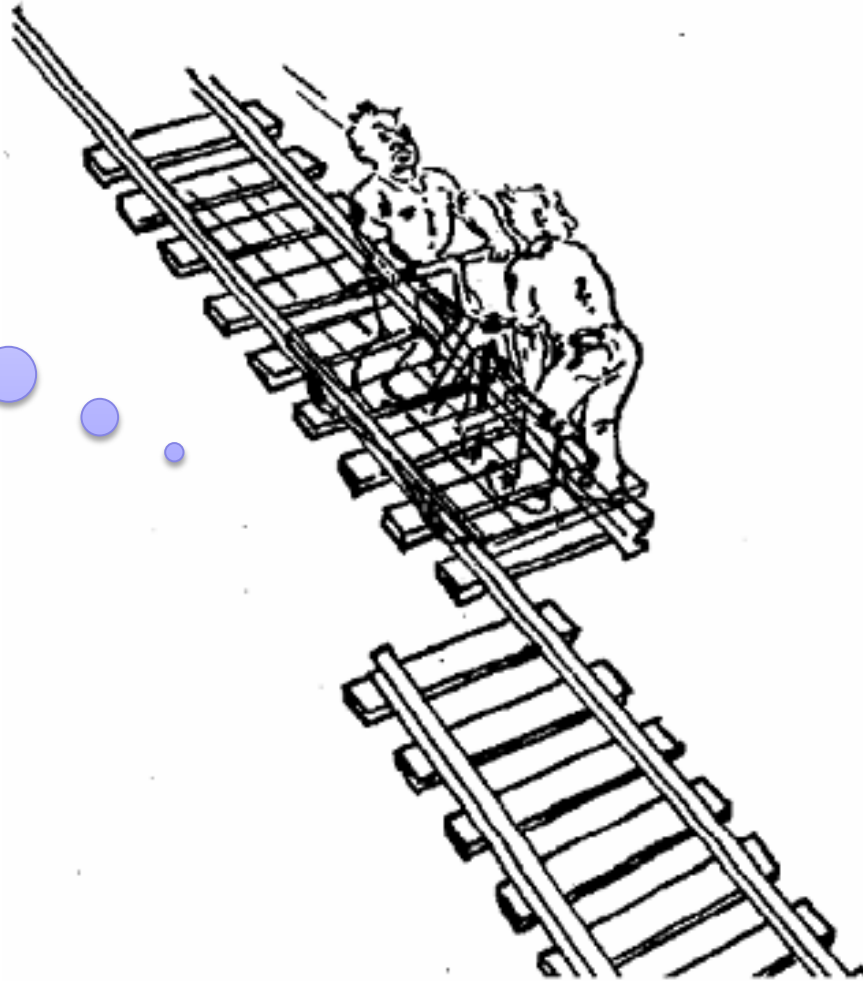
A login form with a yellow background. It contains two input fields: the top one is labeled '用户名' (Username) and the bottom one is labeled '密码' (Password). Below the input fields is a button labeled '登录' (Login).



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

测试？





# 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况的不了解，在测试时作出过于简单的假设。有些问题，包括对软件的功能、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，都无法完全考虑到。而攻击者有足够的时间进行攻击方法的研究。
- (3) **由于软件需求的变动**。软件规格说明书或设计文档无法一开始就确定下来；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，很多安全问题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

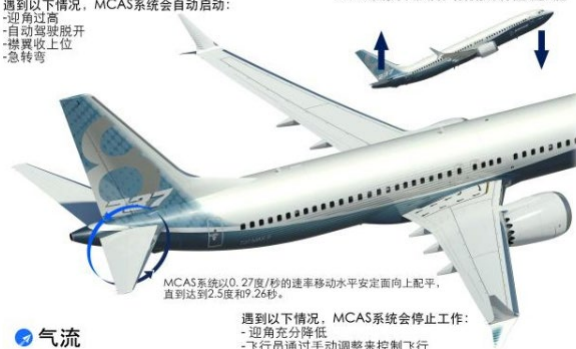
如客户可能在运行软件的过程中，自行安装第三方提供的组件，开发者根本无法知道客户的软件将要和谁交互，软件在运行的过程中出现安全问题。



波音737 Max  
机动特性增强系统 (MCAS)

遇到以下情况，MCAS系统会自动启动：  
- 迎角过高  
- 自动驾驶脱开  
- 襟翼收上位  
- 急转弯

MCAS系统向下推动喷气机头以降低失速风险



气流

遇到以下情况，MCAS系统会停止工作：  
- 迎角充分降低  
- 飞行员通过手动调整来控制飞行



# 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。

❖ **如何考虑软件安全的问题？**

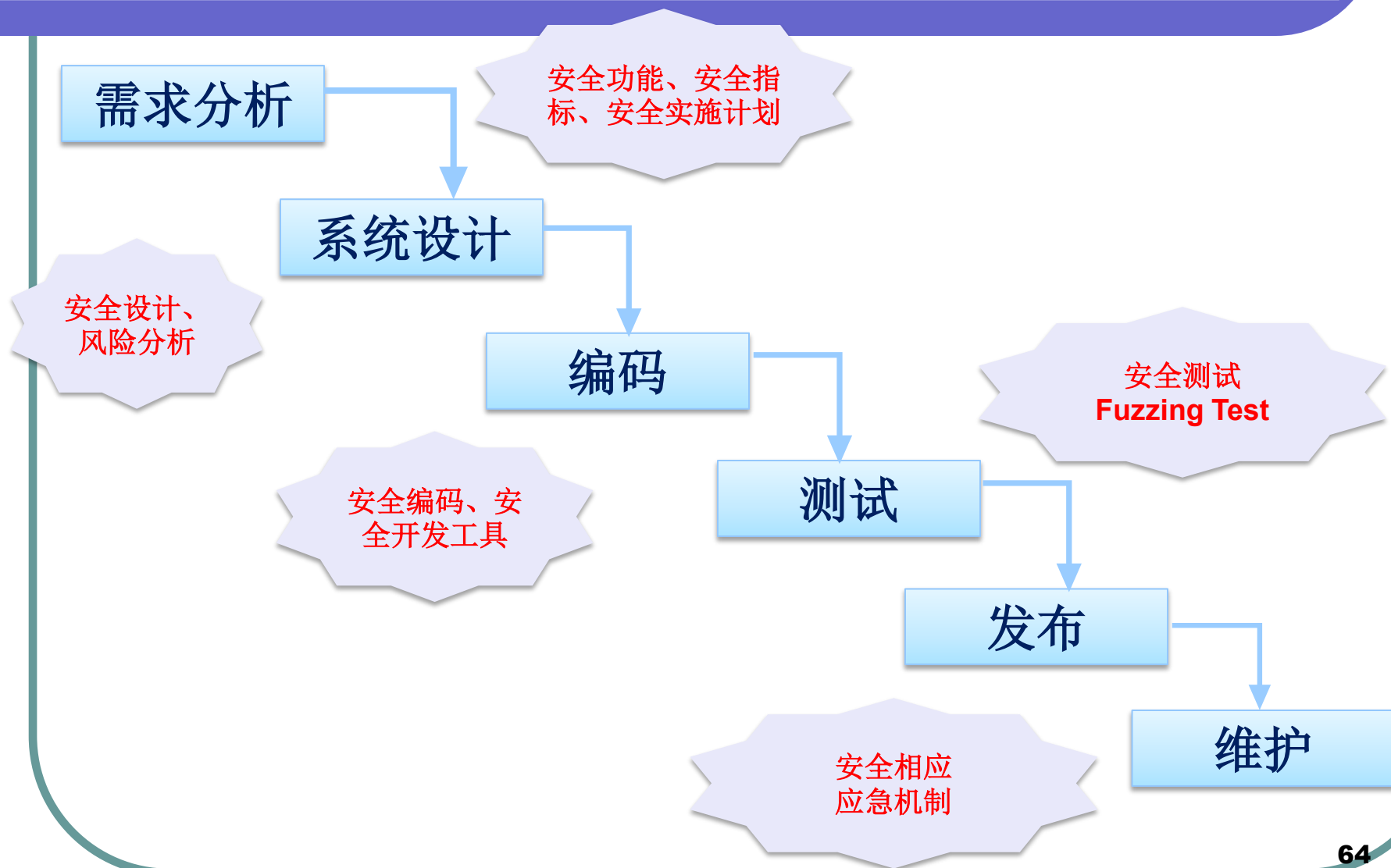
# 软件安全防护手段

- 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**系统

# SDL开发模式

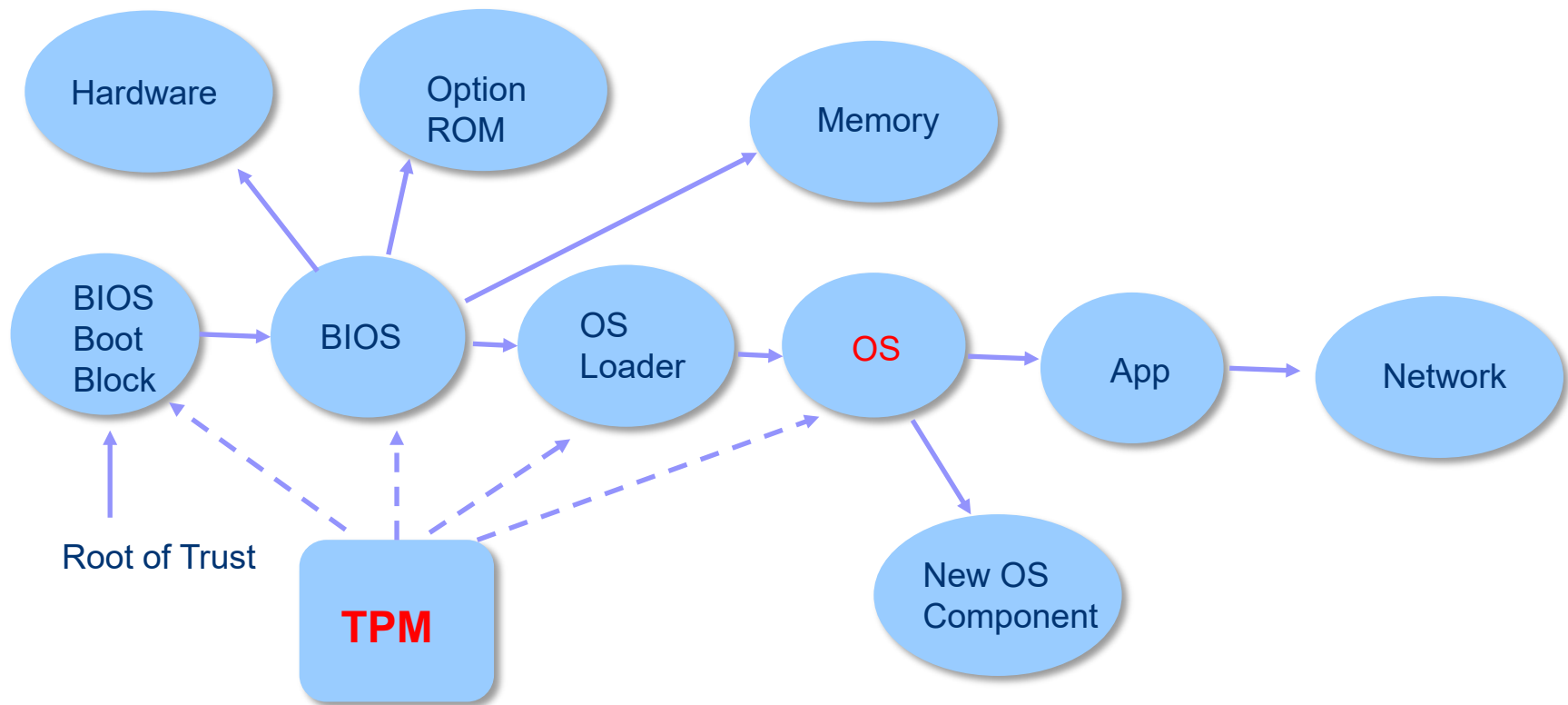




## 2.保障运行环境

- 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

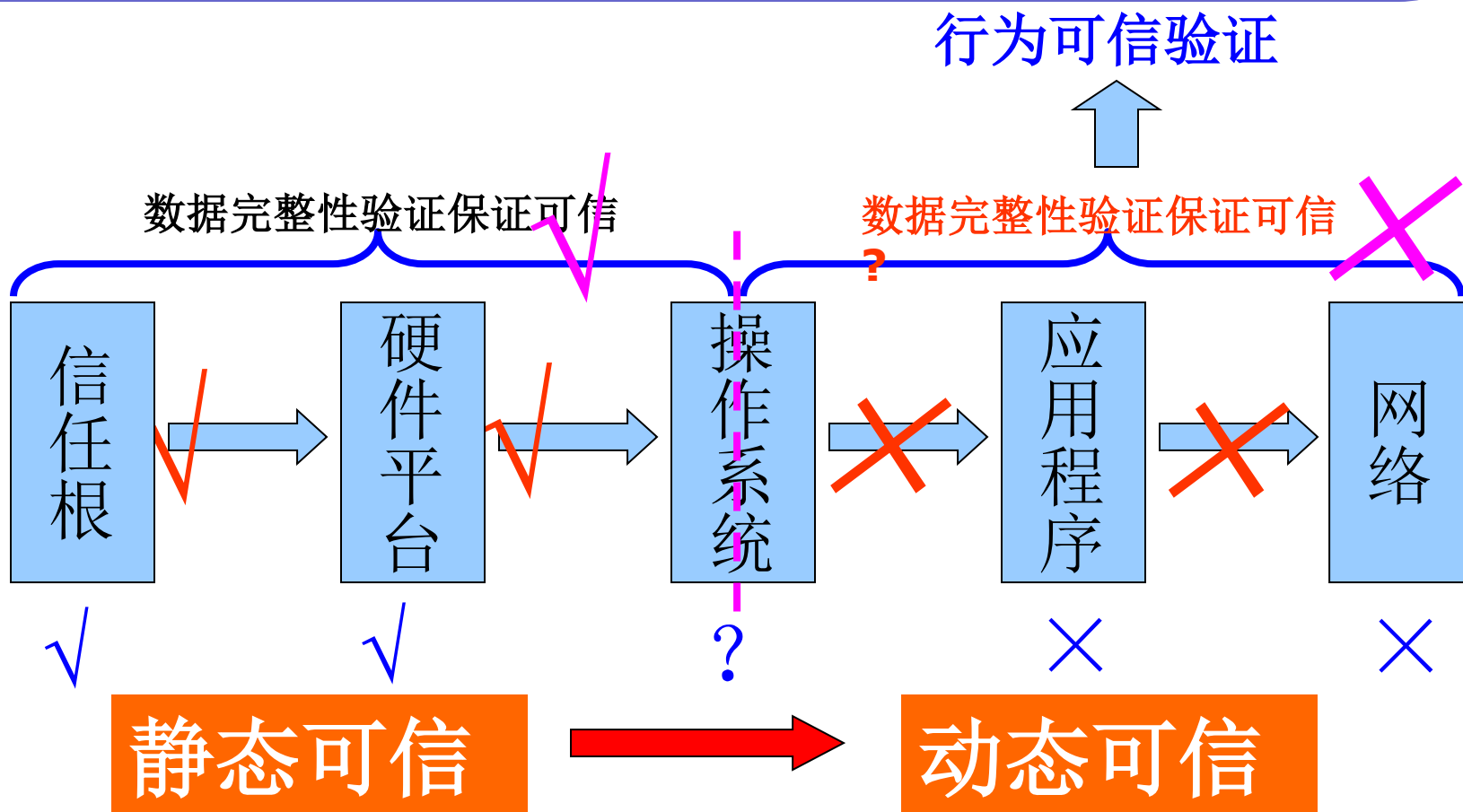


# 3. 加强软件自身行为认证

## □ 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，提供更高效率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目前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SandBoxIE）

### ■ 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？**
- **查阅“美司法部起诉我5名军官—法律文件”（56页，31项罪行），美方有哪些依据？是否合理？**

课后作业在超星，超星的答题将在一周以后关闭，请大家及时提交。

# 精彩内容下章继续...

❖ 第二章见

