

# 软件安全与漏洞分析

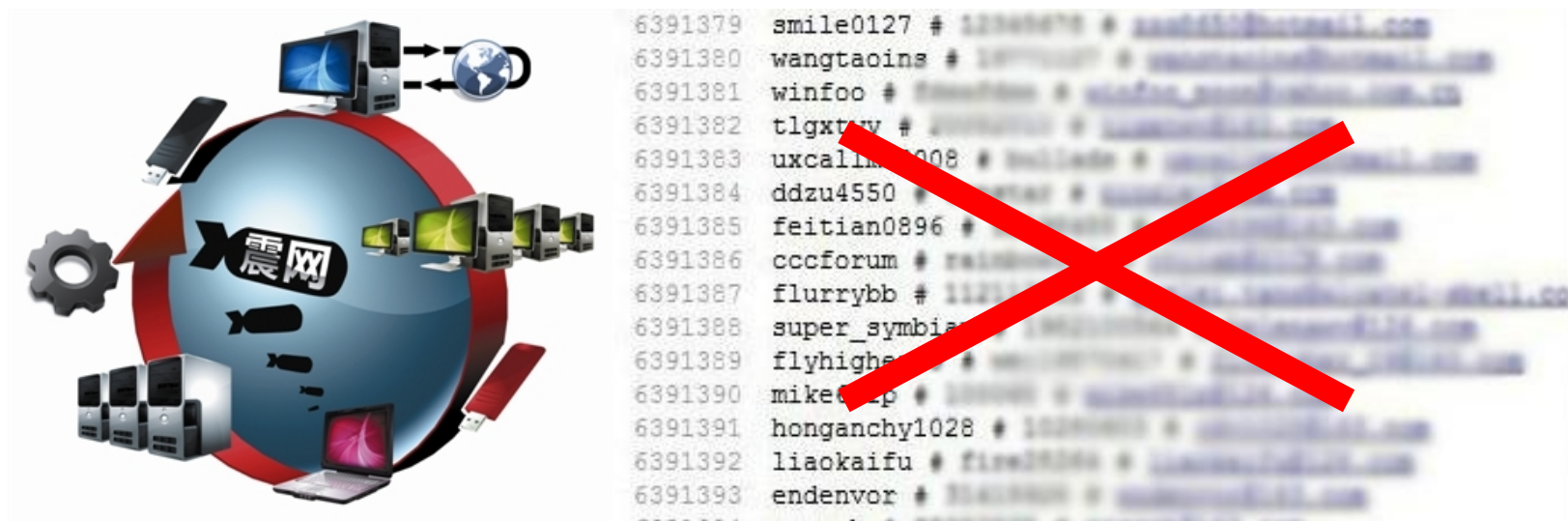
---

## 1.1 在正式开始以前……

# 从这门课程中能够学到什么？

□ 简单地说：

- 了解如何将一个软件/操作系统玩坏；
- 反过来，了解如何避免一个软件/操作系统被玩坏



# 从这门课程中能够学到什么？

---

□ 为此，我们需要学习的内容有：

- 软件安全的范畴是怎样的？
- 软件漏洞有哪些种类？攻击者可以如何利用这些漏洞？
- 恶意代码具有哪些形态？它们如何传播？
- 开发者有可能以怎样的方法实现软件的自我保护？

# 软件安全——from the beginning

# First Computer Bug!

9/9

0800 Antan started  
1000 " stopped - antan ✓  
1300 (032) MP-AC ~~2.130476415~~ 2.130476415  
033) PRO 2 2.130476415  
conv 2.130676415  
Relays 6-2 in 033 failed special speed test  
in relay 10,000 test.  
Relays changed  
1100 Started Cosine Tape (Sine check)  
1525 Started Multi-Adder Test.

1545



Relay #70 Panel F  
(moth) in relay.

First actual case of bug being found.

Grace Hopper  
Photo Courtesy of Hagley Museum and Library



# 软件安全——from the beginning

- 人们很快意识到：和任何其他机器一样，计算机也会出错，并且致命程度还极有可能更胜一筹



# 软件安全——from the beginning

---

## □ 早期漏洞：逻辑设计中的无心之失

- 整数变量的溢出（千年虫）；
- 被忽略的细微误差（宰赫兰导弹事故）
- 未受重视的进制转换错误（火星气候探测者号、阿丽亚娜5型“烟花”）

## □ 转折点：莫里斯蠕虫

# 软件安全——from the beginning

---

“Since the end of Cold War, a lot of former physicist and mathematicians decided to apply their skills not on Cold War technologies, but on financial markets, to create -- as Warren Buffet said -- a different Weapon of Mass Destruction”

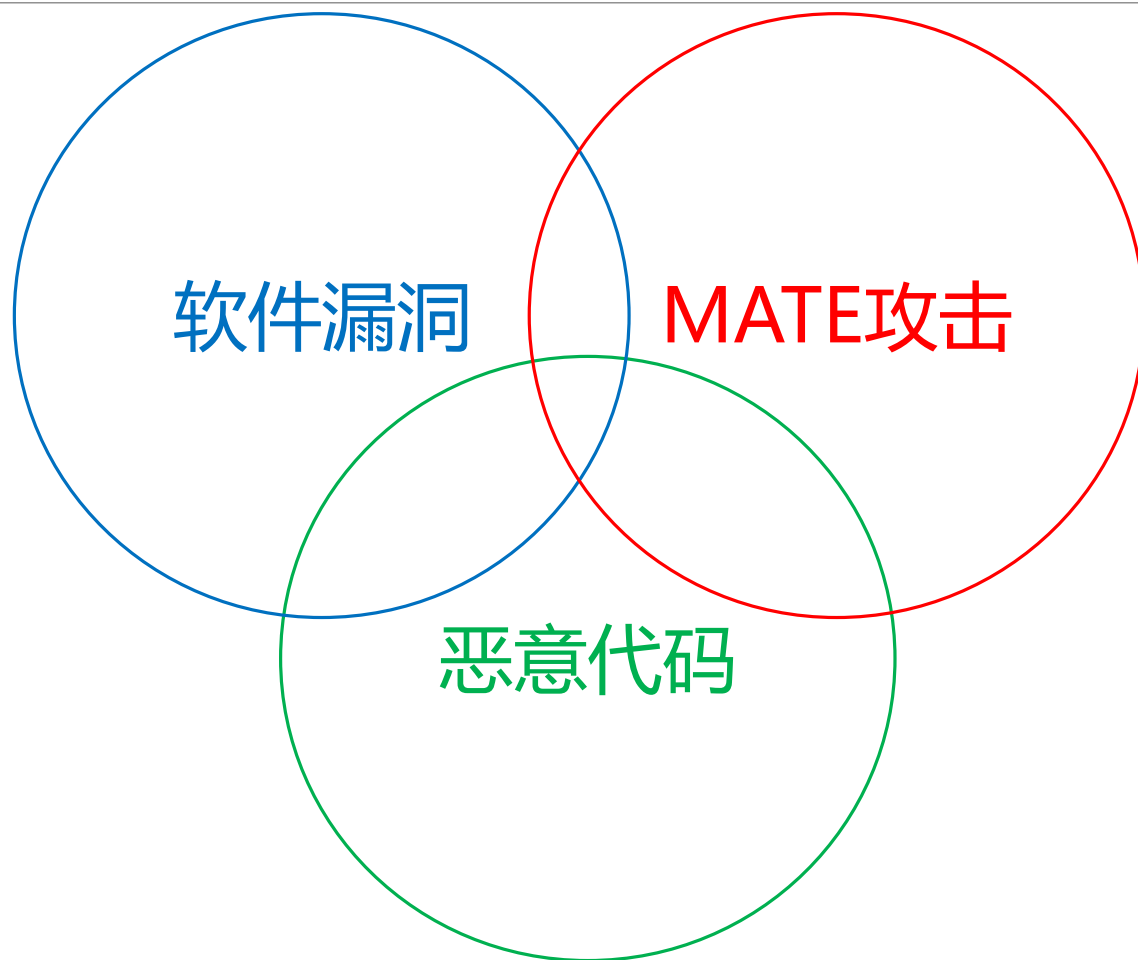
—— 《监守自盗》

金融界并非DALAO们的唯一出路.....

- 黑客攻击使全球网络经济每年损失超4000亿美元（2014）
- 我国“黑客产业链”的规模估计已超过百亿元（2011）
- “僵尸网络”是如今信息安全领域的重要问题之一

# 软件安全——威胁何在？

---





# 软件安全——软件漏洞

- 本质上，程序猿是“语言学家”



可能出错的事总会出错 —— 墨菲定律



# 软件安全——软件漏洞

---

## □ 你可能根本就“想错了” .....

- 例：浮点数的精度始终很高吗？

设为最大值(1024)

尾数末尾精度为 $1*2^{1024-52}$

0/1	11位阶码	52位尾数
-----	-------	-------

与此同时，绝对值最小的规约浮点数的值为 $1*2^{-1023}$

## □ 你的逻辑有幸“完美无瑕”，然而你“翻译”错了.....

- `#include <iostream.h>` 还是“`iostream.h`”？
- 我有一句“`use namespace std`” 不知当用不当用？

# 软件安全——软件漏洞

---

□ 违反规则，不当地使用资源的话……



□ 如果被非法使用的是内存空间的话，会怎样呢……

# 软件安全——漏洞的发现和利用

---

- 错误存在不可怕，可怕的是它们的不为人知



# 软件安全——漏洞的发现和利用

□ 如何确保计算机系统中的漏洞不会伤害到我们？

- 找一个具有30英尺厚墙体的钢筋混凝土地下室；
- 把计算机放进去；
- 这个地下室最好没有门；
- 显然，计算机绝对不能被连接到网络；
- 最好，再切断它的电源……



干得漂亮……但这毫无意义……

# 软件安全——漏洞的发现和利用

- 通用型软件系统最为尴尬的一个难题

用户正常地使用软件系统

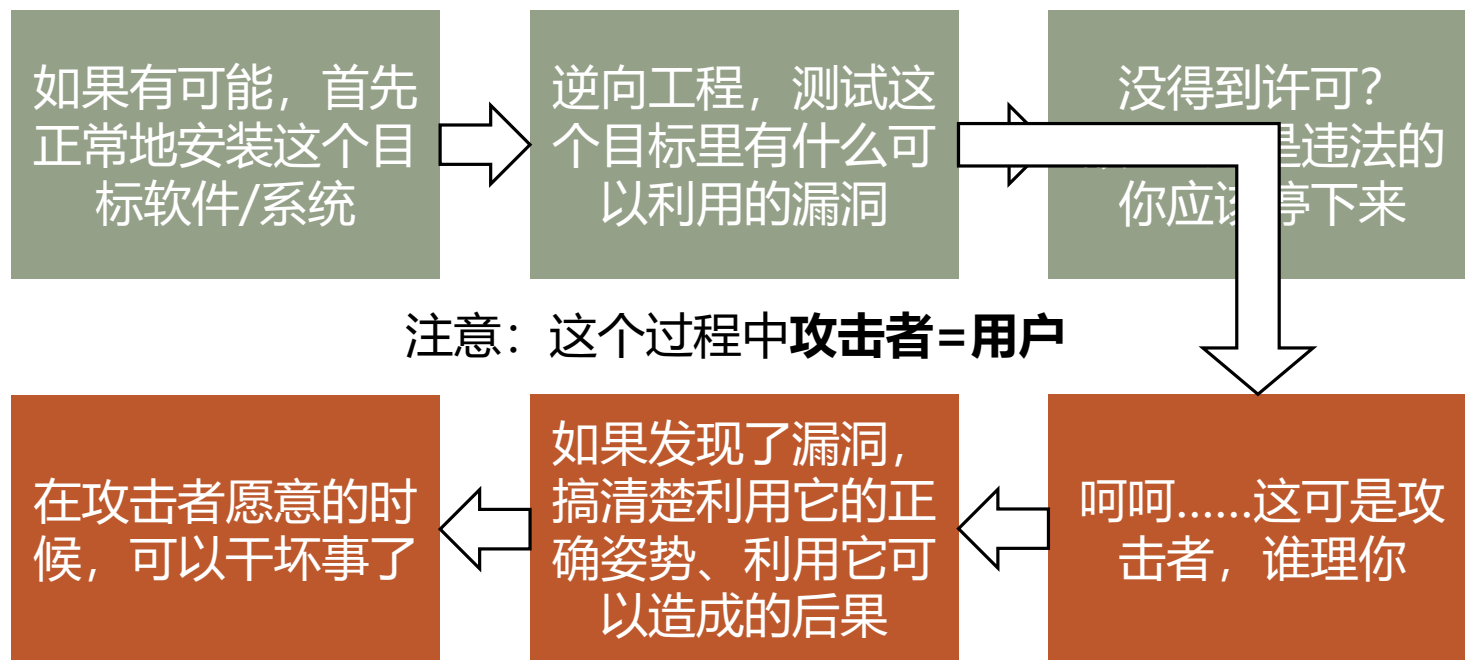
攻击者非法地探查软件系统

那么，怎样确保  
“用户≠攻击者”  
永远成立？



# 软件安全——漏洞的发现和利用

- 当攻击者准备攻击一个软件或系统的时候：



- 这个过程于是（仿效“中间人攻击”）得名**Man-At-The-End**

# 软件安全——漏洞利用的后果

□ 攻击者都可以干什么样的坏事呢？—— 恶意代码！

- 下马 - 拒绝服务
- 中马 - 数据泄露
- 上马 - 任意执行



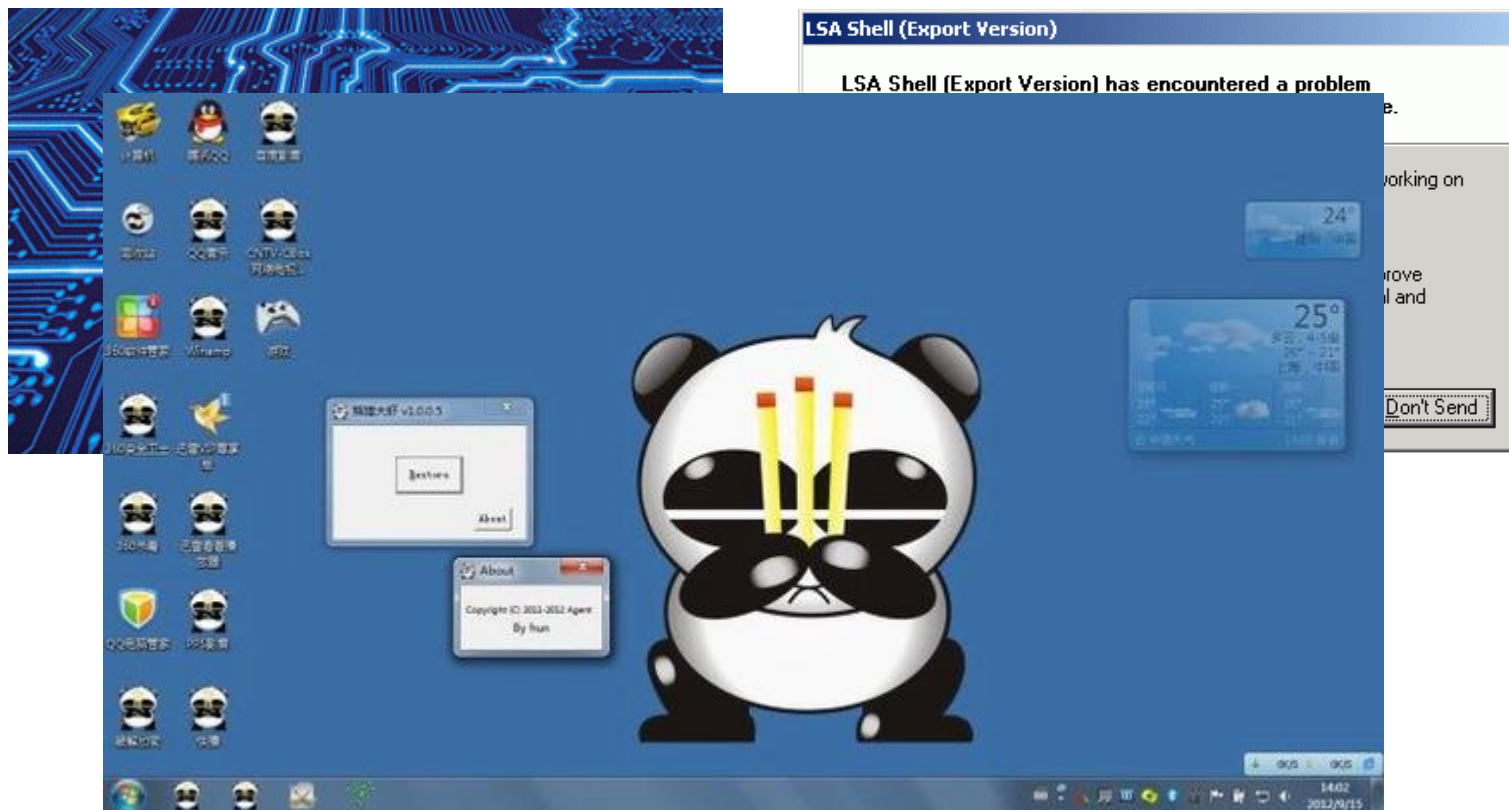
这锅  
不得不背.....

PS：莫里斯蠕虫重复感染同一系统并最终导致宕机的行为，是Dalao一不小心弄出来的“unintended consequence”——即，这也是个bug



# 软件安全——漏洞利用的后果

□ 当然，莫里斯Dalao的后继者们要凶残的多：

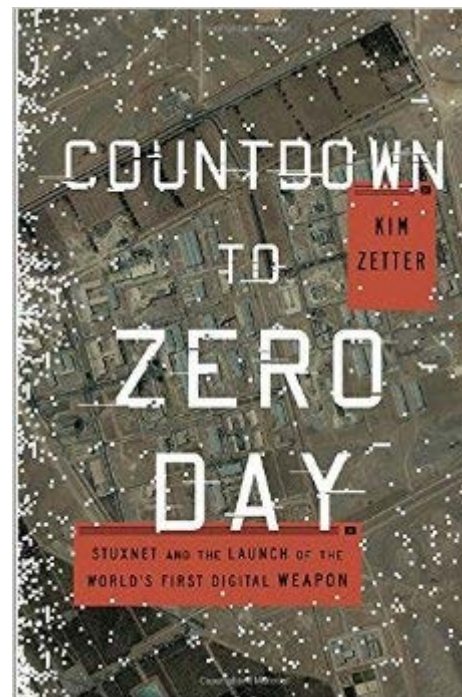


# 软件安全威胁的进化

---

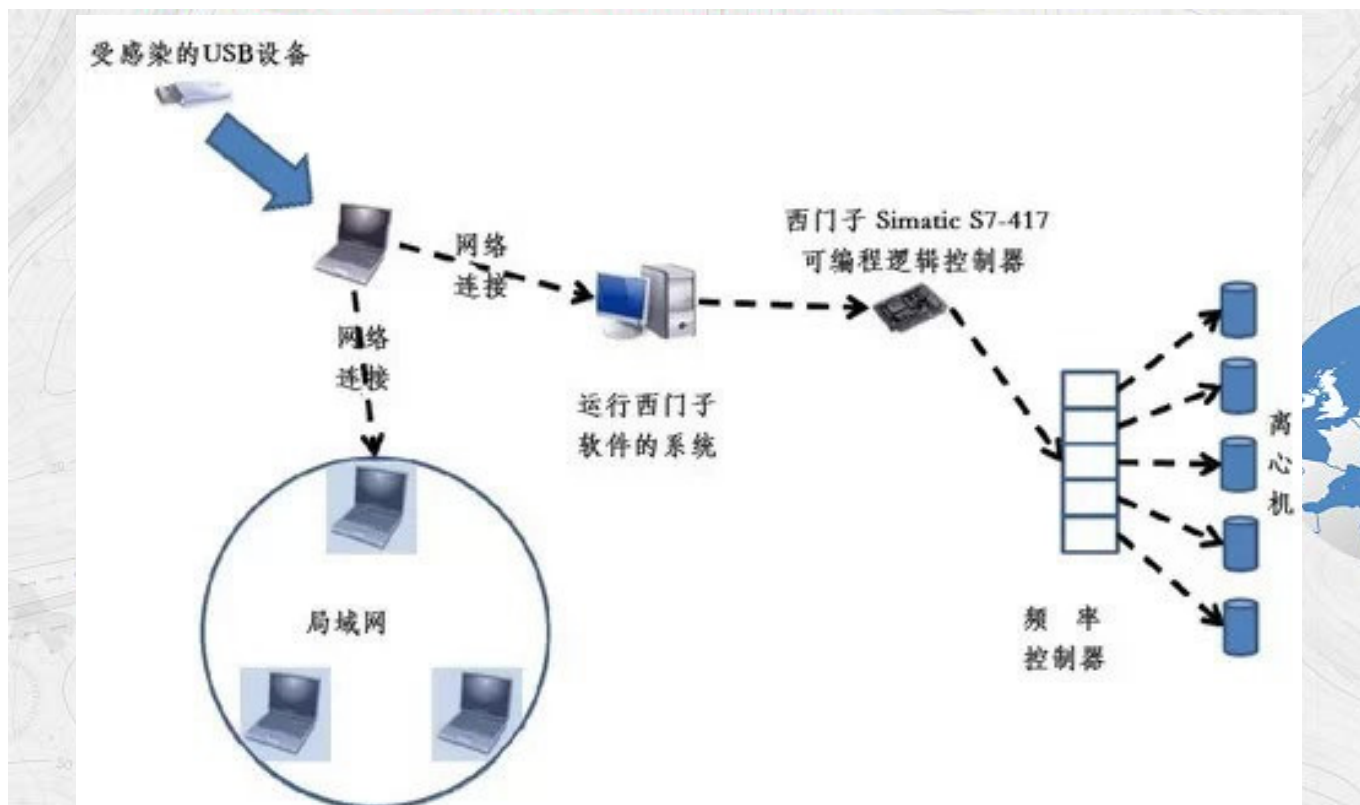
- 当攻击者获得近乎无限的资源：高级持续性威胁(Advanced Persistent Threat, APT)
  - 隐匿自己
  - 针对特定对象
  - 长期、有计划、有组织地实施

震网：近年来的经典APT案例



# 软件安全威胁的进化

## □ “震网” 概览



# 软件安全威胁的进化

---

- 攻击者的身份：个人→组织→国家机器
  - 对待软件漏洞的态度 -- 应该公布的缺陷→值得保密的战略资源
  - 目标 -- 个人电脑→企业组织内部系统→工控系统
  - 威胁 -- 私人主体的经济损失→国家级的战略利益损害
- 威胁在进化，软件安全技术的重要性因而也在与日俱增

# What's next?

---

## □ 课程线索

- 计算机和操作系统基本原理回顾（软件是如何工作的）
- 软件漏洞及其利用
- 恶意代码
- 软件自我保护技术

## □ 教材和参考书目

- 《漏洞战争：软件漏洞分析精要》 《软件安全》（彭国军著）
- 《软件加密与解密》（仅建议参考）