
Introducing Computer Security

What is computer security?

- Most developers and operators are concerned with **correctness**: achieving desired behavior
  - A working banking web site, word processor, blog, …
- Security is concerned with ***preventing*** **<u>un</u>desired behavior**
  - Considers an enemy/opponent/hacker/adversary who is *actively and maliciously* trying to *circumvent* any protective measures you put in place

Kinds of undesired behavior

- Stealing information: **~~confidentiality~~**
  - Corporate secrets (product plans, source code, …)
  - Personal information (credit card numbers, SSNs, …)
- Modifying information or functionality: **~~integrity~~**
  - Installing unwanted software (spyware, botnet client, …)
  - Destroying records (accounts, logs, plans, …)
- Denying access: **~~availability~~**
  - Unable to purchase products
  - Unable to access banking information

Significant security breaches

- **RSA**, March 2011
  - stole tokens that permitted subsequent compromise of customers using RSA SecureID devices
- **Adobe**, October 2013
  - stole source code, 130 million customer records (including passwords)
- **Target**, November 2013
  - stole around 40 million credit and debit cards
- … and many others!

Defects and Vulnerabilities

- Many breaches begin by exploiting a **vulnerability**
  - This is a *security-relevant* **software defect** that can be **exploited** to effect an undesired behavior
- A software **defect** is present when the software behaves incorrectly, i.e., it fails to meet its requirements
- Defects occur in the software's *design* and its *implementation*
  - A **flaw** is a defect in the design
  - A **bug** is a defect in the implementation

Example: RSA 2011 breach

- Exploited an Adobe Flash player vulnerability

1. A **carefully crafted Flash program**, when run by the vulnerable Flash player, allows the **attacker to execute arbitrary code** on the running machine
2. This program could be **embedded in an Excel spreadsheet**, and run automatically when the spreadsheet is opened
3. The spreadsheet could be attached to an **e-mail masquerading to be from a trusted party** (*spear phishing*)

Considering **Correctness**

- The Flash vulnerability is an implementation **bug**
  - All software is buggy. So what?
- A normal user never sees most bugs, or works around them
  - Most (post-deployment) bugs due to rare feature interactions or failure to handle edge cases
- Assessment: Would be too expensive to fix every bug before deploying
  - So companies only fix the ones most likely to affect normal users

Considering **Security**

Key difference:

An adversary is not a normal user!

- The **adversary will actively attempt to find defects** in rare feature interactions and edge cases
  - For a typical user, (accidentally) finding a bug will result in a crash, which he will now try to avoid
  - An adversary will work to find a bug and exploit it to achieve his goals

To ensure security, we must

***eliminate bugs** and **design flaws**, and/or*

*make them **harder to exploit***