Advanced Unix System Administration

Lecture 14
March 15, 2007

Steven Luo
<sluo+decal@OCF.Berkeley.EDU>

Principles of Security

- Know what you're securing!
  - Without an idea of what you need to protect, you're not going to get very far
  - Know what the system needs to do and what the threats against it are
- Security is a process, not a product
  - Can't just put together a good design and rubber-stamp it “secure” – threats evolve
  - Reassess your system's security periodically

Principles of Security

- Keep your users in mind
  - You're doing yourself no good if you make a system so draconian your users look for ways to bypass it
    - If ultra-long passwords = sticky notes on monitor, you might want to look into other solutions
  - Work with your users to determine what they're willing to put up with, and design systems that are useful for them
    - This might require some creative thinking

Principles of Security

- Minimize your attack surface
  - The notional “attack surface” consists of all the possible points of attack on the system
  - By reducing this, you make the attacker's job more difficult – and more importantly, make your job easier
- Implement multiple layers of security
  - Force the attacker to figure out more
  - Give yourself a better chance of detecting intrusions

Principles of Security

- Compartmentalize
  - Creates smaller, simpler parts that are easier to understand and maintain
  - Limits the scope of a compromise of one component (provided you follow other recommendations)
- Minimize privilege
  - Give each component and user only as many rights it needs, no more
  - Reduce the impact of a compromise

Principles of Security

- Assume the rest of the world is hostile
  - Input could come from malicious users or compromised components
  - Environment can be manipulated by attackers
- Implement monitoring and accountability
  - Allows you to identify a break-in or flaws before they become major problems
  - Allows you to track down who/what was responsible

Principles of Security

- The cost-benefit tradeoff
  - There is such a thing as too much security
    - If your security measures cost more than the cost of recovering from a compromise, you have too much security
  - Consider your need for security and the cost of your measures before you start locking down your system

Classifying Vulnerabilities

- Two standard measures of classification
  - Where an attacker needs to be: “local system”, “local network”, “remote”
  - What the attacker can do: “denial of service”, “information disclosure”, “information modification”, “privilege escalation”, “arbitrary code execution”, “system access”
  - Note that the standard classifications usually relate to default configurations – which may not apply on your system

Types of Attacks

- The buffer overflow
  - Many programs use fixed-size buffers to store strings
  - Where string lengths aren't handled correctly, the attacker may be able to write to other memory
  - Could lead to a crash – or to more subtle bugs
    - What is possible is controlled by the memory layout

Types of Attacks

- Integer overflow
  - Integers aren't arbitrary-precision like they are in the abstract!
  - Not accounting for the wraparound behavior of a variable can lead to nasty bugs
  - Problem compounded by differing behaviors of signed and unsigned variables