Advanced Unix System Administration

Lecture 15
April 14, 2008

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

Applications

- DHCP
  - Used for autoconfiguration of clients
    - Client broadcasts DHCPDISCOVER to network
    - DHCP servers reply with DHCPOFFERs containing IP address offers
    - Client broadcasts DHCPREQUEST with address of server whose offer was accepted
    - Server sends DHCPACK to acknowledge the lease and give other config information, or DHCPNAK to reject the request
    - Client sends DHCPRELEASE when done with the address

Applications

- SSL/TLS
  - Used for securing other application protocols
  - Client connects to server, two negotiate a cipher
  - Server sends back a certificate with a key
  - This key is used to negotiate a session key
  - Rest of the traffic in the session is encrypted with the session key

Applications

- HTTP
  - Developed to transfer web pages, now used for general file transfer and other purposes
  - One of many text-based protocols
  - Request syntax:
    - [command] [parameters] [HTTP version]
    - Additional headers separated by \r\n and ended by two \r\n
  - Reply syntax:
    - [HTTP version] [status code]
    - Headers, two \r\n, data

Applications

- HTTP con't
  - Common requests: GET, HEAD, POST
  - Reply codes: 200 (OK), 302 (Moved), 403 (Forbidden), 404 (Not Found), 500 (Internal Server Error), 503 (Service Unavailable)
- FTP
  - Text-based control protocol, but more complicated
  - Active mode: client connects to port 21 and gives a port for server to send it data on

Applications

- FTP con't
  - Passive mode: client connects to port 21, server sends random port for client to connect to receive data on

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