A Reality Check

- Can we prove a single server secure?
- No dearth of mechanisms
  - Permissions for files, programs
  - SSL, IPSEC, Kerberos
  - JVM, .NET CLR Security
  - Web configurations, Firewall rules, …
- No one technique can prove everything
  - But, we have made advances on important parts
- Key challenge: usable security

The Microsoft View: secure by default

- Use safe programming languages/platforms
  - Prevent (not detect) buffer overruns
  - Use fine grained permissions controls in code
- Use declarative security configurations
  - When possible, don't do security in code
  - Out of the box security by default
  - Readable XML config files for DRM/File Permissions/Network/Web Server/CLR
- Use public standards and protocols
  - Every deployment is a mix of Windows/Linux/Solaris/…
  - Just interoperability is not enough, security standards must be vetted and well understood

My View: theory has come of age

- We can all do Needham-Shroeder
  - Using dozens of provers: FDR, Proverif, MSR,…
- We can also do sophisticated protocols
  - automated analyses of: JFK, SSL, Web Services
- Now, on to full, composite architectures
  - Maybe even a Windows Server ☺
- Key Challenges:
  - Prove concrete protocols, not just abstractions
    - Model full messages, real configs, running code
  - Develop compositional proofs
    - System is secure even with bad clients, servers, other protocols running in parallel