

# Securing Services Orientated Architecture



Andrew van der Stock  
OWASP

# What is the problem

- [ SOA: Buzz word du jour
- [ Be part of the rush to web-enable legacy code
- [ Not a lot of security advice - just a huge “do it!” from integration vendors anxious to relieve you of dollars

# Why you'd still do it

- [ Straight through processing
- [ Increase the value proposition to your customers
- [ Reduce the cost of developing that Web 2.0 fancy all singing all dancing front end

# Architecture Violation

- [ There is a reason why the n-th tier is deeply buried in your network.
  - The code sucks.
- [ How do you prevent XML injections in 1980 vintage COBOL?
- [ Are the implicit security decisions made in 1980 still valid?
  - Probably not.

# Access control

- [ SOA is only safe when you know WHO, WHAT, and WHEN something is happening
- [ Many SOA toolkits use WS-Security to do the “WHO” component
  - This requires a PKI and proper credential management
  - Bite the bullet - it's time

# Validation

- [ Validation is still required
- [ Old code doesn't validate for modern attacks
- [ If your SOA toolkit supports XSD - use it
- [ If your SOA toolkit does not support validation, you will need to write it yourself... or choose another toolkit

# State Management

- [ SOA is perfect for “Take \$100 from my account”
- [ SOA sucks when asked to do anything more complex
- [ You must be able to manage state securely - relying upon client-side state is not a good idea

# Auditing

- [ Need non-repudiation for STP
- [ T&C's
- [ Understand your legal obligations

# Injection attacks

- [ Business rule attacks
- [ XML injections - particularly if the old code is not aware
- [ LDAP injections for X.509 certificates - be careful about enrolment

# Reporting

- [ PDF files can contain buffer overflow attacks
- [ Use virus scan on all files received and sent to users
- [ Be careful about using static URLs for reports

# Conclusion

- [ SOA has a part to play in the enterprise
- [ Know who can call you, and how
- [ Be careful of the transactions you allow out into the big bad world
- [ Don't forget the lessons of the last 30 years



Questions?