Secure Web Services Architecture

A Case Study

Matthew G. Marsh

President, Paktronix Systems LLC

Chief Scientist, NEbraskaCERT

Overview - Section 1

- Web Services
  - What is it
  - Why is it
  - Who cares
- Traditional n-Tier Web Services
  - MultiTier Arches
- Network Security
  - Theory
  - Reality
  - n-Tier
  - General
  - Tips

Overview - Section 2

- Detailed Analysis of Case Study
  - Systems
    - Multiple Secure Environments
    - Least Privilege
  - Network
    - Protocols
  - Management
    - Updates / Upgrades
- Q & A & HandsOn

S1.1: Web Services

NEbraskaCERT

- What is it
  - Services you get over the Web
  - Commonly defined WRT Programming Paradigm
    - JAVA, XML, SOAP, and so on
- Why is it
  - Good buzzword for sales pitches
  - Extension of traditional environment to "Distributed Systems"
  - You don’t understand - Mr. Sales Puppet does!
- Who cares
  - "The Boss" read about it in an airline magazine
  - So now you care

S1.1: WebServ – Part Deux

NEbraskaCERT

- Data Representation
  - What is the definition of data
- Intercommunication
  - How do I manipulate data
- Description
  - What does my data look like
- Discovery
  - Where is my data

S1.1: WebServ Styles - .1

- XML, SOAP, WSDL, UDDI
  - XML – eXtensible Markup Language
    - Same as SGML only k3w1r
  - SOAP – Simple Object Access Protocol
    - Uh-huh – Remember ASN.1 …
  - WSDL – Web Services Description Language
    - The XML way to say “Web Site”
  - UDDI – Universal Description Discovery Integration
    - X.500 is simple by comparison

S1.1: WebServ Styles - .2

- REST - REpresentational State Transfer
  - Remember GOPHER protocol (Archie/Veronica)
- What made the WWW take off back in 1991
  - URL – Uniform Resource Location
  - HTTP – HyperText Transfer Protocol

The core of the WWW is the combination of a global resource location scheme using DNS (URI/URL) with a simple and easy resource consumption mechanism (HTTP).

S1.1: WebServ Styles – 0xff

- Consider how you as graphical consumer know how the web page you are looking at was created
- No peeking at the source!!
- Hmmm – no clue – eh?
- Was it static or dynamic?
- How do you tell?

The internal representation of a resource is IRRELEVANT!

So why do you want to know what type of Web Services are operational within any given system?

S1.1: The Answer

NEbraskaCERT

Because you want to violate something!

After all – Security is just a cost center

S1.2: Tier Environments

- Web Services must run only in a n-tier environment
  - That is pronounced “ahn – tear” “ehn-virulent”
- Traditional Structure of Networks & Applications
  - Network: All People Seem To Need Diet Pepsi
  - Application: People Admire Darwin
- Location in both respects is key
  - Which System defines which name
- In both cases any Tier may consist of many parts
  - Especially true in Application Tier
- Logical Design has Defined Tiers
- Physical Design has one or more server(s) per Tier
- Physical vs. Logical is very important when defining Network Management capacities

S1.2: Traditional 3-Tier

NEbraskaCERT

Presentation (WWW Server)

- Some studies refer to the “Client” tier
- Considering the Client as the “Presentation”
  - –JavaScript, HTML, XML

Application (CGI, J2EE, Cobol)

- Not necessarily an independent server
- Best defined by Usage
  - –Applications ~= Programs

Data (dBaseIV, SQL, Contacts.TXT)

- Should not imply a DataBase in the operational sense
- Best considered as referential

S1.2: n-Tier Concepts

- Presentation Tier
  - "Front Facing" systems WRT Client connectivity
  - Greatest Exposure and Visibility of WebServ System
- Application Tier
  - Multi-Tier depending on Application Program paradigm
  - May not be amenable to Clusters, Load Balancers, etc
  - “Glue Logic” design structure important for control
  - Heavy Interdependence on Presentation and Data
- Data Tier
  - Defined by the operational data characteristics
  - Optimization may depend on use
- Interconnectivity
  - Protocols, Protocols, Protocols
  - Security, Security, Security
  - Oh yeah – Management

S1.2: More Concepts

- n-Tier Architecture
  - Traditional separation of processing duty.
  - Similar to the concept of an exploded mainframe
    - Presentation (Green Screen)
    - Processing (COBOL)
    - DataBase (oh yuck – pick your own horror…)
- But since this is “exploded” we can actually obtain access to the points in between
- Even better we can slip in and reside within the middle or back systems
- Consider the difference between a SOAP procedure to index your DB and Melissa LovinU…
- Personally I can wait to see the first SOAP virii…

S1.2: General APPSec

NEbraskaCERT

- Protection Mechanisms
  - Document your software
    - Yes – this means UML and Data Flow Diagrams
      - Unified Modeling Language
  - Good Programming and Design Practices
  - Respect GIGO
  - Leverage the Synergy of Parallelistic Realities
    - Ummm – y'know – use _lots_ of Snort probes…
  - Consider the simplest representation of the data
    - AND USE IT
  - Try to constrain data type flow
    - XML in – XML out
  - Understand the systematic structure
  - Strive for ISN or at least respect PoI

S1.3: NetSec - Theory

- CIA – Confidentiality, Integrity, Accessibility
  - Confidentiality
    - Proper protection and use of business data
  - Integrity
    - Proper application and operation of business rules
  - Accessibility
    - Availability of business infrastructure when and where needed

- Operational Data of physical structure
- Physical Data of logical structure
- Consideration of the entire network as a single system

S1.3: NetSec - Reality

- Define the most important business needs
  - Ex: Product Inventory in Jones Street Store
- Determine critical systems WRT that need
  - Ex: Jones Street Store inventory DB server
- Determine "How" to define system availability
  - Ex: SQL query with a known response
- Define limits of "How"
  - Ex: SQL query returns within 30 seconds
  - Ex: One retry of query is permissible
- Define escalation procedure when "How" fails
  - Ex: Go home and hide under bed

- Network Security is all about Business Continuity

S1.3: n-Tier NetSec

NEbraskaCERT

- SNMP
  - Basic network and system connectivity
    - Ability to trap and monitor system processes
    - Ability to hash and verify files and data
  - Baseline of global structure interaction
- Presentation
  - Use WWW verification tools – “Page There” tools
  - Use form testing utilities to verify interactions
- Application
  - JMX – API for peering within JVM structures
  - I/O verification (form testing / CGI testers)
- Data
  - DB Monitoring tools
  - Custom SQL / Data request testers

S1.3: General NetSec

- Traditional Network Management
  - SysAdmin and O.S. specific performance tools
  - SNMP and "Frameworks"
  - MVC structure (Model – View – Controller)
- Alerts, Monitors, and Correlation
  - Alerts define problem situations
    - May be predefined or WRT ongoing system(s) operation
    - Issued by problem area or system
    - Usually Boolean in terms of issuance
  - Monitors
    - “Polling” for predefined data and/or operational structures
    - Data Types usually predefined
  - Correlation
    - Combining monitor and alert structures for insight
    - Commonly spoken of as "Performance Metrics" or "Security"
- Ideally Platform and Network Agnostic

S1.3: NetSec - Tips

NEbraskaCERT

- SNMP
  - Use IPX where possible
  - Use Version 3 with full authPriv and Inform traps
    - Separate passphrases for auth and Priv
- Serial Logging
  - AKA Out Of Band (OOB) Logging
  - Especially useful for NID/HID systems
- Time Synchronisation
  - Does not need to be accurate merely precise
- Read Only DASD
  - Especially useful for static content
  - Works well with well behaved programs (Apache)
  - Read Only NFS is tolerable
- SSH / SSL

This is The