Building an Enterprise Infrastructure to Securely Manage Access to Web Applications

Paul Munsen – IT Architect
Navistar, Inc.
IIT Netsecure 09
March 12, 2009
About Navistar

• World class commercial truck and engine manufacturer
• Sells products through approximately 300 dealerships across North America – most of which are independently owned and operated.
• Currently expanding into additional markets and regions across the globe
Problems to Solve:

• How do we provide secure access to web based applications from our dealerships?
• How do we allow users to log in to all applications with a single user ID and password?
• How do we allow traveling employees to access web sites when at customer locations?
Potential Solutions:

• VPN?
  – Good for transport layer security
  – Not easy to identify individual users

• Application specific security
  – Good for some applications
  – Saves on licensing costs
  – Difficult to manage across multiple applications and development groups
  – Very difficult to use with 3rd party applications (ERP, Portals, etc…)

• Web access management product – reverse proxy
Web Application proxy

• Provides consistent single-sign on interface
• Provides transport layer security (SSL)
• No software needs to be installed on web servers or client PC’s
• Allows for consistent access control mechanism
• Web applications only
• Not 100% transparent
Benefits of Proxy Architecture

• Proxy architecture increases security by not allowing direct user access to web servers

Source: IBM Redbook SG246014
Navistar before use of Web Application Proxy

• About a dozen different dealer-facing web based applications
• Some required their own user ID’s and passwords
• Most applications used Windows authentication
• All applications required custom user authentication/authorization code
• No applications were available over the Internet – private network connections were required
Technology in use at Navistar

• Wide variety of web technologies in use:
  – Microsoft IIS/ Active Server Pages
  – Java/ Tomcat
  – Java/ IBM WebSphere Application Server
  – CGI-based systems
  – Mainframe, Windows, HP/UX, VAX, AS/400, etc…
Solution Implementation

- Decided to use IBM’s Tivoli Access Manager – similar products are available from other vendors
- Customized system to authenticate against Windows to be compatible with majority of existing applications
- Built Network DMZ to host Internet-accessible proxy servers

Source: IBM Redbook: SG246014
Building a scalable architecture

• Initial rollout – Jan 1, 2000
  – 1 application
  – 1000 named users
  – 3000 web hits per day
  – Availability: Approximately 98%
• Current environment – March 2009
  – Approximately 200 applications
    • 55,000 named users – 15,000 unique users will log in on a particular weekday
    • 11 million web hits per weekday
    • Availability: Approximately 99.95% (about 20 minutes of downtime per month)
Infrastructure

- Internet User
- Load Balancer
- Reverse Proxy
- Web Servers (IIS, Apache, etc...)
- Internal User
- Reverse Proxy (multiple)
- LDAP Directory (multiple servers)
- Policy Server (management console)
- Externally Hosted Web Servers
High Availability

• Load balancers in front of most components
• Multiple LDAP Servers
  – Using peer to peer replication
• Multiple proxy servers for high availability and to handle load
  – Proxy servers managed from a single management point
• Proxy servers can provide load balancing to provide high availability for application web servers
• Monitoring
  – Availability
  – Performance
  – Trending for capacity management
Application Integration

• Configuring an application to go through proxy
  – Set up a “junction” or connection between proxy and web server
• Standalone Site: http://server1/index.html
• Accessed through proxy
  – https://proxyserver/junction/index.html
Application Integration

• Things to consider
  – Security
    • Transport layer (SSL or HTTP)
    • Mutual Authentication of servers
    • Propagation of user’s identity (user ID) to application
    • Passing of additional identity information
Application Integration

• Things to consider (continued):
  – Integration
    • Can the application code be modified
    • Do you trust the application?
    • What configuration options are available within the application?
    • Will this application run on multiple servers or a single server?
Passing Identity Information to Applications

• Benefits to externalizing and centralizing identity information
• Extended attributes sent in HTTP header
  – User Name
  – Account Numbers
  – Any other relevant information that may be shared across applications
• Particularly useful when integrating with applications hosted by outside providers
• Easy to retrieve attributes sent in HTTP header
  • String userID = request.getHeader("iv-user");
Future Directions - Federation

- Federation is a mechanism by which users can achieve single sign-on across companies using a standards-based protocol such as SAML.
Questions or Comments

- Contact information:
  - Paul.Munsen@navistar.com