Honeywell

ExxonMobil Research and Engineering

Role Based Access Control for Industrial Automation and Control Systems

Johan B. Nye
ExxonMobil Research and Engineering Co.

Kevin P. Staggs
Honeywell ACS Advanced Technology Labs

27 October 2010

abstract

There has been a lot of focus on the security of Industrial Automation and Control Systems (IACS) from the perspective of network security and device security. There has been some focus on user security configurations of IACS. This paper and presentation will address the need for implementing Role Based Access Control (RBAC) in an IACS.

First we will describe the design objectives and what we mean by RBAC. Then we will define the various roles that are needed for configuration, management and usage of an IACS, and show how to define those roles in a *Windows operating system* domain environment.

We will also show how to configure the roles consistently in a multi vendor IACS environment. Guidelines will be given for configuration of the *Windows operating system* domain and how to integrate the multiple vendors' systems in the same domain. In addition we will address requirements that need to be placed on the IACS vendors to facilitate easier integration into an RBAC environment.

presentation topics

- design objectives
- basic roles for Industrial Automation and Control Systems (IACS)
- common issues and challenges
- *Windows operating system* Active Directory design for IACS
  - reference IACS architecture
  - Forest/Domain design
  - Organization Unit (OU) design
  - Accounts design
  - Groups design
  - Group Policy Object (GPO) design
- system/application design for IACS
  - system design requirements
  - secure by design
  - easy integration into user domains
  - secure when not in domain
  - integration of *Windows operating system* and IACS system/application security

design objectives

- enhance the safety, reliability and security of industrial control and automation systems
  - Integrity, Availability, Confidentiality
  - no reliance on Enterprise IT systems for essential services
  - no single points of failure between manufacturing locations and business units
- individual accountability
  - a means to authenticate an individual user
  - confirm that individual users are assigned the least privilege for their role
  - a means to determine the actions of an individual user
- minimize support and maintenance costs
  - minimize the number of Domains
  - integrate multiple vendors' products in the same Domain
  - minimize the number of "authentication databases"
  - minimize the effort to implement security (secure by default)
- essential services for the IACS
  - Level 1 real-time control, including peer-to-peer control
  - operator view and change of process values
  - operator notification of process alarms

basic roles for Industrial Automation and Control Systems

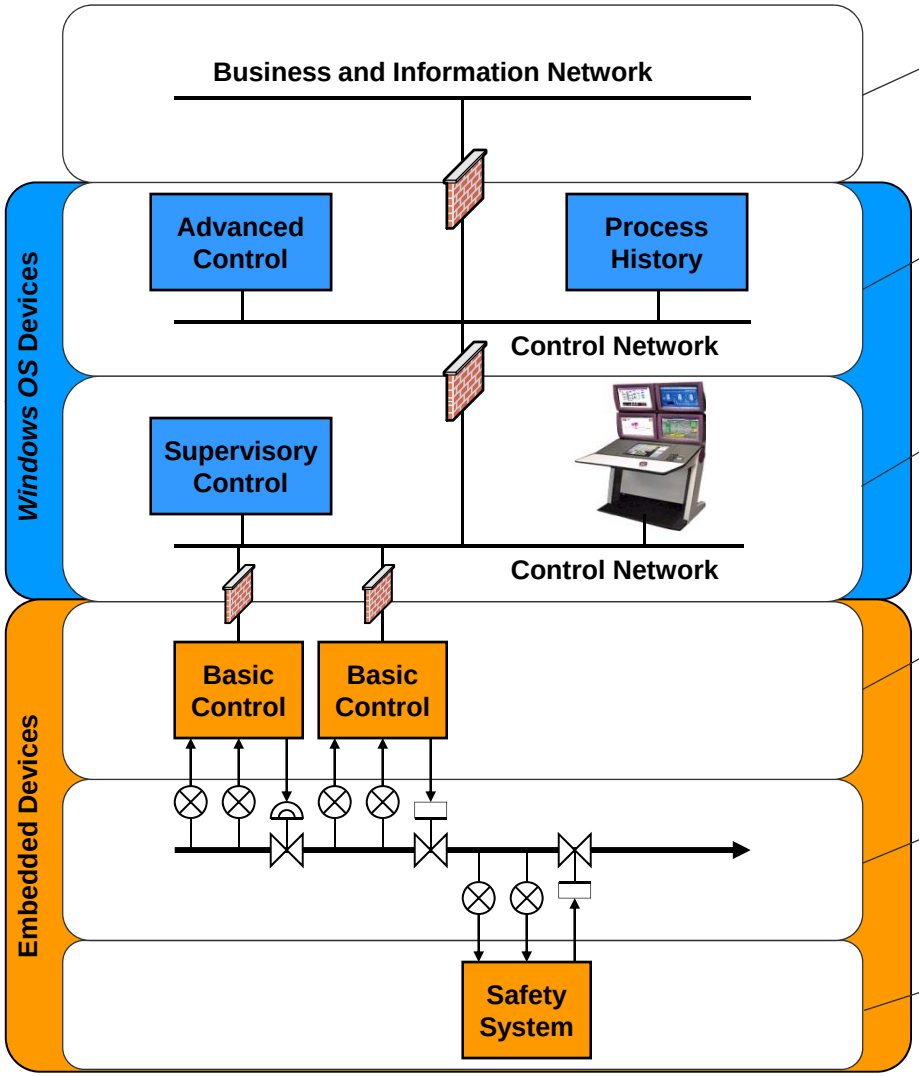
- minimum set of roles for all IACS
  - View only
  - Operator
  - Control Application Engineer
  - Control System Administrator
- additional roles to consider
  - Operations Supervisor
  - Instrument (Field Device) Technician
- characteristics of roles
  - least privilege
  - mutually exclusive privileges (separation of duties)
  - a single person can have multiple roles
  - *but a single role cannot be split between different users*
- best in class system/application
  - vendor provides a basic set of roles and privileges
  - asset owner can modify or create roles and privileges

common issues and challenges

- roles that are not adequately defined or segregated
  - lack of view only role
  - engineer and administrator roles combined
- roles that are not adequately tested
  - each role must be individually tested
- roles that have escalated privileges
  - e.g. normal operational accounts with administrator privilege
- passwords that cannot be changed
  - especially for essential service accounts
- use of anonymous accounts and groups with open membership
  - accounts: Guest, IUSR_machine,
  - groups: Everyone, Authenticated Users
- lack of single sign-on capability
  - multiple authentications needed (e.g. *Windows operating system* Domain and vendor's application)
- security models that don't share the Domain
  - e.g. GPO applied at Domain root

Windows operating system Active Directory design for IACS

reference IACS architecture



Level 4 – Business Planning and Logistics

Level 3 – Manufacturing Operations and Advanced Control

Level 2 – Operator Interface and Supervisory Control

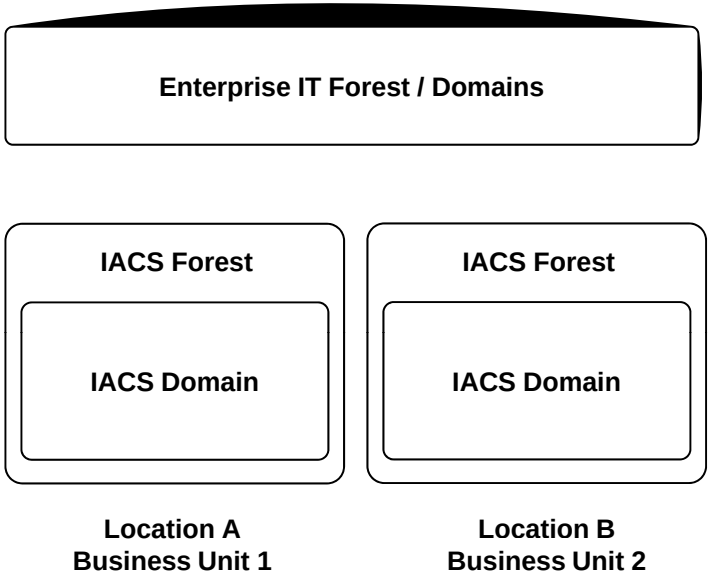
Level 1 – Basic Control

Level 0 – Sensors and Actuators

Safety – Safety Instrumented Systems

Forest/Domain design

- single *Windows operating system* IACS Forest for each manufacturing location and business unit
  - ensures no common-cause faults between manufacturing locations and business units (e.g. *Windows operating system* Forest schema)
- minimize the number of IACS Domains in each IACS Forest
  - typically one Domain per Forest
  - reduces administration and maintenance costs
- no trusts between Enterprise IT Domains and IACS Domains
  - reduces security vulnerabilities in the IACS Domain associated with the use of Everyone and Authenticated Users groups



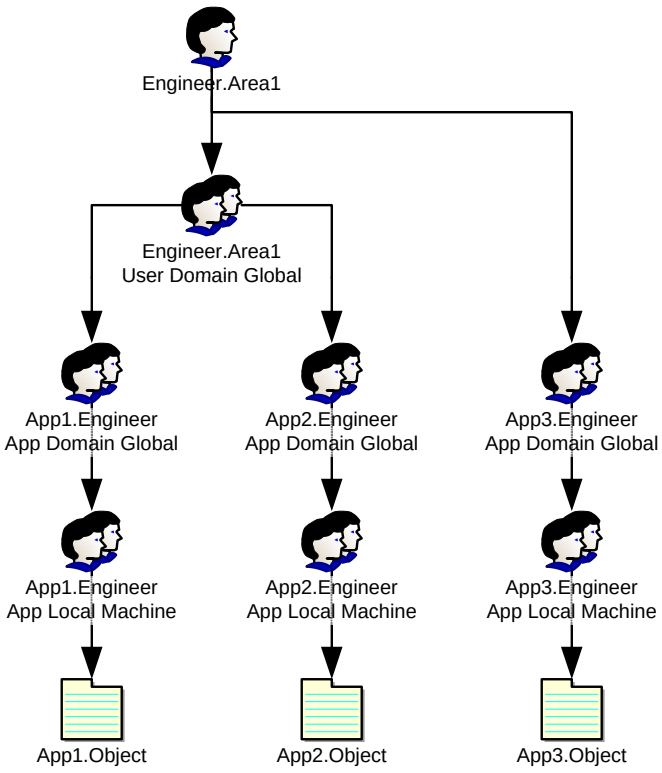
Organization Unit (OU) design

- reasons to create an OU
  - a different group policy is required for the user or computer accounts
  - to delegate administrative authority of a portion of the domain to a different organization at the manufacturing location
  - a functional OU is required (e.g. for a specific vendor's control system)
  - the creation of an OU would reduce administrative or support costs
- OU design principles
  - separate accounts from computers
  - separate groups from accounts
  - separate different "security zones"
    - e.g L2 and L3 computers
  - separate vendor-specific systems/applications
- performance considerations
  - maximum depth of OU structure <5 for adequate performance

- **Accounts**
  - **Groups**
    - **Administration**
    - **Remote**
    - **Services**
    - **Standard**
    - ***Application Specific (as needed)***
  - **Users**
    - **Administrators**
    - **Console**
    - **Remote**
    - **Services**
    - **Standard**
- **Admin**
- **Builtin**
- **Computer** *(empty)*
- **Domain Controllers**
- **ForeignSecurityPrinciples**
- **LostAndFound**
- **Shared Resources**
  - **Printers**
  - **Shared Drives**
- **Systems**
- **Process Control Systems**
  - ***Vendor-specific (as needed)***
- **Process Monitoring Systems**
  - ***Vendor-specific (as needed)***
- **L3 Servers**
  - **Remote Servers**
  - **Standard Servers**
  - ***Application Specific (as needed)***
- **L3 Workstations**
- **User** *(empty)*

Groups design

- Types of groups
  - Universal Groups
  - Domain Global Groups
  - Domain Local Groups
  - Local Machine Groups
- basic approach
  - Objects are assigned permissions through Local Machine groups for each role
  - Application Domain Global Groups are placed into Application Local Machine Groups for each role
  - User Domain Global Groups are placed into Application Domain Global Groups for each role and scope
  - Individual user accounts are placed into User Domain Global Groups or Application Domain Global Groups for each role and scope
    - tradeoff between ease of admin vs troubleshooting
  - avoid the use of Universal Groups and Domain Local Groups
  - minimum granularity of scope is the “operator console” area



Group Policy Object (GPO) design

- basic approach
  - separate Computer policy and User policy
    - Computer policy (HKEY_LOCAL_MACHINE)
    - User policy (HKEY_CURRENT_USER)
  - separate vendor standard GPOs and asset owner changes to this GPO
    - vendor GPO → asset owner GPO
- linking GPOs to OUs
  - Computer GPOs linked to Computer OUs
  - User GPOs linked to Accounts OU and filtered using Groups
  - occasionally User and Computer GPOs linked to the same OU
    - when user policy depends on computer type (e.g. remote access)
  - avoid Block Inheritance and No Override for easier troubleshooting
- key learnings
  - minimize use of multiple GPOs per object
    - rule of thumb maximum 4 GPOs per object
  - avoid applying vendor GPOs to Domain root
  - avoid linking GPOs to the Admin container
    - contains admin accounts of “last resort”
    - Block Inheritance to Admin container

Accounts design

- types of accounts
  - essential accounts
    - console accounts (operator role)
    - service account for essential functions
  - non-essential accounts
    - view only
    - engineers
    - other users
  - privileged accounts
    - administrators

system/application design for IACS

system design requirements

- build basic set of roles into the system
  - minimum set of roles predefined for operation, configuration and administration
- do not depend on administrator role for normal operation
  - no function for control should ever require administrator rights
- consider database security as part of the design
- use built-in security capability of the host system
  - integrate host system security into process application security
- easily integrate into existing process control domains
- minimize the amount of security configuration necessary for the end user
  - secure by default
- integrated security for *Windows operating system* and process applications

secure by design

- utilize *Windows operating system* security functionality
- avoid the use of built-in users and groups for IACS applications
  - avoid the use of default installed operating system users or groups
    - exception is administration role where administrator is needed
  - install IACS-specific security for IACS applications
    - create and install IACS-specific groups as local groups
      - one local group for each IACS role
    - secure applications (file system and registry) to IACS local groups
- determine database roles required
  - don't use default database roles
    - database administrator role should only be required for initial creation
  - define IACS-specific database roles
    - make sure there are multiple roles
- assess installed *Windows operating system* functions needed for each role
  - prevent usage where not needed by a specific role
    - use group policy where policies exist
      - provide Group Policy Object to end user
- minimize usage of local accounts
  - should only be required for essential accounts

easy integration into user domains

- provide Group Policy Objects to install into user domain
  - separate user policies from computer policies
  - do not install by default at the root domain level
- define all system local groups and their intended role
  - provide instructions or templates to make domain groups members of local groups
  - domain groups may already exist for many customers
  - provide creation for domain groups where they don't already exist
- minimize domain security configuration required
  - minimum user requirement should be creation of user accounts and assignment of those accounts into the proper domain groups

secure when not in a domain

- security using local groups allows default security for workgroup based systems
  - group policy security is not applied
  - administration is duplicated on all nodes
  - account and password synchronization required across all nodes
  - difficult to change password, especially online

integration of *Windows operating system* and IACS system/application security

- *Windows operating system* security protects the file system
  - Authentication, Authorization, Auditing (AAA) functionality built in
- IACS system/application security provides security for process assets
  - many provide separate AAA functionality
- use *Windows operating system* authentication for both file system security and IACS system/applications security
  - authentication should be integrated with IACS application
- use *Windows operating system* credentials for IACS system/application authorization
  - IACS authorization can be tied to *Windows operating system* users or groups
    - assign IACS access levels to *Windows operating system* users or groups

summary

- covered role based access control with least privilege design principles
- covered *Windows operating system* Active Directory design principles
- demonstrated how to integrated multiple systems/applications in the same Domain
- demonstrated how to implement single sign-on for both *Windows operating system* and systems/applications

secure by design, secure by default, secure in operation

Windows is a registered trademark of Microsoft Corporation in the United States and other countries. This is an independent publication and is not affiliated with, nor has it been authorized, sponsored, or otherwise approved by Microsoft Corporation.