Chad Ballay

**WEEK 7**

|  |  |  |
| --- | --- | --- |
| **Access Control Method** | **Implementation Method(s)** | **How It Benefits Multilayered Access Control** |
| User Access Control Profiles | Role Based Access control | Sets the base default access that people should have to do the role they are assigned. Minimum access needed feedback loops can be put into place here. |
| Systems Access | System Access Control Lists | System wide security policies for activities that occur. (What type of events should trigger what type of metric generation.) |
| Applications Access | Account privileges within application. Application permissions for execution | Finer grained control is possible on what users can have access to specific functionalities of the application or even if they can run the application itself. |
| File and Folder Access | Read/Write access controls for user and group level | Limits access to underlying data and configurations for applications. Facilitates segregation of duties into roles. |
| Data Access | API gateways, Database connections, etc… | User, System, Application, File & Folder access is limited first but in the end the data is where the objective is. This is the closest to the actual target and therefore this is the last line of defense. Subverting all the others are just means to an end. |

Every organization’s infrastructure, goals, org chart and compliance needs are in flux over time. New people are hired, change job duties, leave the company. New technologies are onboarded, expanded in their usage, sunsetted. Laws come into scope, get interpreted, etc…. What all these events mean is that we can’t design a perfect system that’ll handle any and all changes that will be needed in the future. By the time you define the explicit inputs/actions/outputs of that process flow, the landscape will have changed. The best we can do is to implement a series of access control methods that give mechanisms for conceptualizing access and controlling access. Communicate the policies and then define the standards of how the work should be done.

Abstracting the individual user into a series of job duties so we can define generic functions they can do is the core starting point. Defining what they can do and what they shouldn’t be able to do flows outward to then defining what “tools” they can use to do that job. Then defining the modalities that these pairings of role/tool interactions on the data substrate of your business processes is the final layer. Who is doing the work, how are they doing the work, what is the work that is being done. None of this is a permanent view but putting things into this conceptual framework allows you to build a approach that can be built/refined upon over time.