**EMCS2000 Introduction to Computer Security**

Assignment: Your Organization’s Web Environment Security

Brian Russel Davis, [brian\_davis@brown.edu](mailto:brian_davis@brown.edu)

NASA, to my knowledge, does not deploy complex web applications for public consumption. Public facing websites are purely informational and don’t have interfaces that accept UGC. I have heard that certain divisions offer access to large bodies of scientific data, but I am far removed from those activities, so I don’t know how security for these activities is managed. I mainly interact with internal users building applications for internal use only. Web applications are used internally for a number of scientific and administrative purposes. Security ( for the most part ) follows the [Discretionary Access Control Model](https://en.wikipedia.org/wiki/Discretionary_access_control) ( DAC ). Once employees are given a machine they don’t have access to anything non-public without explicit access from the owner of that resource. Furthermore, employees and contractors are not allowed to share documents unless they have been explicitly granted permission to share said documents with others. Even if you are hired as a NASA employee or contractor you start your journey with access to just about nothing ( not even a laptop ). Access to every asset, website, and resource must be granted with a request.

As noted above, access to internal resources requires a formal request. This includes every internal web application, internal website, physical access to buildings or rooms and all the group level implementations of Sharepoint which contain documents. The formal request includes non-editable information about the training that is required to access the resource, the level of internal security confidence that is necessary, citizenship requirements and a record of the requestors standing when it comes to each of these elements. For an example, a request to have admin access ( for using the  **sudo**  command in the terminal for example ) on a NASA issued laptop requires training on the responsibilities of having admin rights, US citizenship, a certain level of security risk confidence, and approval from a sponsor ( usually a NASA civil servant ) and the manager of the resource ( in this case technical support ). Once access is given the resource usually requires authentication each time the resource is accessed. This authentication is almost always verified with a Smart Card that is attached to the computer and guarded with a PIN. Furthermore, if authentication fails more than few times, an account will be locked until the user calls the NASA help desk and verifies their identity.

As you can see, the process of getting access to things is pretty slow and cumbersome, but it serves a purpose. Since NASA employs people from nearly every country on the planet, and the overall security vetting process only qualifies an employee for public trust level information, the security strategy is akin to letting most everyone in “the house” but keeping the most of the “doors” inside the house locked. For the most part, this makes sense, however in my module 5 report, I proposed that DAC remain in place, but the process of granting access be automated using machine learning:

*“Role-Based Access Controls, Attribute Based Access Controls, and Lattice-Based Access Controls all give us some notion of control over an aspect of systems that users interact with.   
  
In* ***RBAC*** *hierarchy is represented by the sets U, R, P, S (users, roles, permissions, and sessions, respectively) and defined by these base formulas.  
• UA ⊆ U × R (user assignment)  
• PA ⊆ R × P (permission assignment)  
• RH ⊆ R × R is a partial order called the role hierarchy or role dominance relation written as ≤.****ABAC*** *is very similar but switches the concept of roles with attributes.****LBAC*** *popular among engineering who create REST Services and API’s describes security “in terms of the lattice (a partial order set) where each object and subject have a greatest lower bound (meet) and least upper bound (join) of access rights.”   
  
Knowing this, how do the traditional models calculate access and usage? For example, two nuclear scientists are working on some of the same projects, with the same level of security clearance, for the same organization. However, they may or may not be permitted to see the same files in a given directory if some of the documents include discussions with military intelligence about HEARTBEAT, for example. Even among people with the same level of security clearance, information about certain topics may be privileged on a need to know basis. Using traditional RBAC having files with varying access levels in the same folder would create a complex security policy conundrum, one where a special policy is attached to the document or a restriction where a document cannot be included in a folder with other documents. Using ABAC to filter the documents would mean creating and assigning some special attributes only accessible to one of the scientist. With highly sensitive documents this still might work, but as we mentioned previously people rarely take the time to describe access and usage at a granular level, per document. Using LBAC to filter the documents would mean assigning the document to a security class within certain bounds for one scientist and out of the bounds for the other, a class that might not exist outside of blocking information about one topic.   
  
By using machine learning to perform curation on data objects which feeds a dynamic policy engine, a document with sensitive conversations between one nuclear scientist and military intelligence about HEARTBEAT, can be hidden from the other nuclear scientist based on the content in the document not on a policy attached to a folder, a predetermined uncommon attribute or a complex set of security bounds. The curation engine recognizes the conversation as privileged, prompts the user for guidance, and feeds the curation into a policy engine that determines access based on the relation of the subject to the document. The rule created in the engine is not transitive to any nuclear scientist with the same clearance, but only those that have a relationship with the topic and a history in the conversation. In a sense the control access threats each piece of content like a sensitive social media post, allowing the user to fine-tune the audience and usage, prompting them with curation ques based on similar bits of content.”*

*“I am going to call what I have described up until this point a* ***Dynamic Relational Trust-Based Access Control ( DRTBAC ).***  *This keeps track of the content that is produced by the user and aids in the curation of content for the use in a real-time policy engine. Much like the AI policy engine in the Chromium project this engine uses the analyzes the content and tries to come up with a policy based on the sensitivity of the content. In many ways, the NASA model has the same controls, but DRTBAC model automates the process and allows for the possibility of security within documents.*

*Implementation of DRTBAC involves installation of a custom agent in NASA HostScan system, SAML integration with Sharepoint for the PIV authentication methods and robust backend to train the machine learning model and retain policy data.” [[1]](#footnote-0)*

1. Davis, Brian. “EMCS Module 5 Report.” *EMCS Module 5 Report*, Brian Russel Davis, docs.google.com/document/d/163jtEVJtVFmKY\_jwQnI8GJb7vHc-TA8kBZo0VK64ayw/edit?usp=sharing. Page 17 [↑](#footnote-ref-0)