Review 2 of NGAC-GOADS,

INCITS Project CS1/2195-D, Rev. 1.3, 11 April 2014

From: Hilary H. Hosmer, President of Data Security Inc. Bedford, MA

Date: June 26, 2014

Here are a few more comments about NGAC-GOADS 2195, now in public review.

OVERALL:

This NGAC-GOADS document (CS1 2196-D dated 11 April 2014) is not nearly as complete or well-written as the GOADS section (clause 6) of the NGAC-FA document (CS1 2194-D dated February 2012). Its goal appears to be to provide a mathematical version of the NGAC standard, but it uses so many mathematical approaches (set theory, logic, BNF, C programming, and Z notation), it may put off all but the most mathematical readers.

- The writing is dense in contrast to the Feb. 2012 document, and no examples clarify meaning until the Annexes. It might have been easier to insert math into the earlier GOADS document, which includes better definitions, and additional facts.

GOALS

The forward says "complete detailed description of the definitions and abstractions needed to realize the NGAC standard." The document is not there yet.

AUDIENCE:

"The GOADS reader is expected to be mathematical, computer-literate and very familiar with the earlier NGAC-FA document." These words should appear in the Abstract as well as in the Foreword and Scope. The thorough reader is expected to know set theory and notation in section 4, logic in

section 5, plus BNF notation and C programming language in Annexes A and B. Z notation is also used.

The primary audience is probably lead software engineers responsible for adopting the NGAC standard for their installation.

ORGANIZATION:

This document meets NIST's conventions for organizing documents. However, it appears to be cobbled together from different works.

- Except for the last sentence, the scope is the scope for the whole NGAC rather than for just this document. This makes it hard to understand what this particular document is all about.

- The use of many different mathematical notations suggests that several sections were borrowed from prior work.

- Why do A and B, which are informative and belong with the main section, appear in an Annex?

- Why does the document end so abruptly?

- The math notation in Appendix D helped a lot!

SCOPE:

Except for the last sentence, this section appears to describe the scope of NGAC rather than the scope of this document. (Another borrowed item?) Suggestions:

Minor

- Sequence of sentences should be reversed: e.g. in para 2, put well-known policies first, then mechanisms. Include mandatory access control (MAC) among well-known policies.

- Para 4, replace first "situation" with "environment.

- Replace "central organization is combined " with "central organization may be combined"

- Para 5, insert "access control" to make: "in addition to its support of access control policies".

Significant

- Where is NGAC control information covered?

- It seems ambitious to go beyond policies to data services, email etc, (para. 5) which are not supported in this document.

- Interoperability is a laudable goal, but it's not clear which elements are necessary to maintain interoperability. (para 6)

SCOPE GRAMMAR:

- para 2, "policies" should be "policy"
- use article "a" in "security model is <u>a</u> formal representation"
- para 3, "adaption" should be "adaptation"
- para 4, "policy policies" should be "policy"

DEFINITIONS 3.1

Consider better names for policy elements. For example, **Policy Types** (e.g. MAC, DAC, RBAC) are more intuitive than "policy classes".

Since "Authorization" has traditionally been associated with user login, passwords, etc. (and still is used that way in NGAC-FA (p. 21), **admit/deny access decision** is an alternate expression for "Authorization decision", whenever "access control decision" is meant.

"Containers" is a counter-intuitive term for user and object attributes. "Roles" in medical RBAC have user categories such as Doctors, Nurses, Administrators, Operators, Technicians. Putting someone or something in a category (e.g. red heads) seems more intuitive to me than putting someone in a "red head container".

It would be difficult to replace any terms at this late date, but policy type makes many policy variations, such as RBAC-medical, or RBAC-retail, or MAC-NATO and MAC-USA, more intuitive as well.

Avoid using terms not yet defined: e.g. "PIP", "configured NGAC policy", "assignment relation", "rooted at the policy element".

Since "configured policy" has not yet been defined, use "pre-defined policy".

Avoid using the same words in definition. E.g. Access right set = a set of access rights…

3.3 **May not** seems more negative than neutral.

3.4 Appendix D not C (para 6)

4. Abstract Data Structures

Forward pointers [] are good.

Using the same **U** symbol for User and set union is confusing (see 4.4.2). Try different fonts.

Simple illustrative examples would help first-time readers ease into the math conventions.

Users U = Alice, Bob, Admin

Access rights AR = read-only, change policy database (PIP)

Processes P = programs initiated by users
Alice -> email; Bob -> wordprocessing

Objects O = database, document1, file2, message 2

Operations OP = (read, write, delete, add, create)

Policy Types PT = (MAC, DAC, NGAC, Chinese Wall)

4.4 Relations need to be explained better. For example:

In mathematics a relation is a file or array containing rows and columns of data. Each column contains one class of data, such as name, telephone number, or date of birth. Each row (called a tuple) contains data related to just one entity, thereby linking a set of elements. NGAC uses each tuple to link policy elements, including users, objects, operations, processes and policy classes.

Five-part tuple

Policy Class	*User*	*Process*	*Object*	*Operations*
DAC	Alice,	email,	message 2,	read, write, delete

Relations, continued.

Under a role-based access control policy RBAC (policy type), a doctor would be assigned to a Department (eg. obstetrics).

The doctor (user) would be linked to all of his or her patient records (objects). He would have permission to read, copy, and add to those records (operations). (Note that he would never have delete permission for medical records, because medical records are never deleted. Corrections are just added.) In an emergency, the doctor could request info about any patient, not just his own.

The doctor or nurse (users) could request medication lists, write prescriptions for patients and schedule future appointments (processes). They could also search for (process) all patients with similar diagnoses (user attributes), or check the side effects of classes of medicines, such as antibiotics (object attributes).

If the Doctor were reassigned elsewhere, the patients and their records would be reassigned to another doctor. (Administrative operation)

4.4.2 Assignment (see NGAC-FA clause 6 for a better description)

Note confusion in defining PE when User and Union have same symbol ∪.

The assignment relation is binary, meaning it has two elements or two sets of elements. Example:

> Assign patient-records-1 to DoctorX.
> Assign DoctorX to Obstetrics Dept

Explain ascendant and descendant plus taking on qualities of descendant.

[Note I skip the math in section 4 because it requires too much time and effort to parse. Not clear it is useful either, compared to NGAC-FA clause]

Minor notes: Do we need names “ars” and “pees”? Page 10. 4.4.4.2 thru 4.4.4.4

4.4.5 grammar; add comma in “and, for those events

Page 13 Question: Does a denial of service threat exist if no other access control request can be satisfied until the obligation request is complete?

p. 16 5.2 Insert after “interpreted as changes to data structures,” or to the contents of data structures, maintained at the PIP

Change order of administrative create policy commands to follow a likely real world scenario. I suggest:

1. policy class
2. a user attribute in a policy class
3. an object in a policy class
4. an object attribute in a policy class
5. a user in an object attribute
6. a user in a user attribute
7. an object attribute in an object attribute

p. 19 The comments are very helpful! Annex A could use something similar.

The many examples up to page 29 are very useful. Then the document stops abruptly, before going on to Annex A. No summary.

p. 30 A.1 Grammar
line 2, change “of” to “or”

p. 30 para 1 “defined obligation” would be clearer as “predefined obligation”. In FA, “contrived relations” would be clearer as predefined relations.

p. 37 Grammar line 2 “enforce” should be “enforced”