Created:June 13, 1994
Modified:August 6, 1996

Gemini Controls Group Report

Security in the Gemini System

Steve Wampler

TN-C-G0038(gscg.sbw.080)/01

This report describes the security issues affecting the Gemini Control System and discusses the mechanisms and policies involved.

1.0 Introduction

Security is a concern in any large software system and the Gemini Control System (GCS) is no exception. This report describes the security issues that affect the GCS and attempts to distinguish between the mechanisms required to ensure a secure system and the policies that should be implemented using those mechanisms.

2.0 Security Issues

When a person mentions ‘security’, there are a number of concepts that are brought to mind, such as:

- Access security-- a secure system should be resistant to deliberate attempts to gain unauthorized access to that system as well as resistant to external actions that may adversely impact the system.
- Abuse security-- a secure system should make it difficult for an authorized user to use the system in an unauthorized manner, whether the abuse is deliberate or accidental.

At the same time, however, a secure system needs to still be usable - the security mechanisms should not interfere with authorized use of the system. A perfectly secure system is also likely to be perfectly unusable. Furthermore, actions that might be considered abuse in one context may be acceptable use in another context. For a telescope system such as Gemini, it is important that security not be made too restrictive, particularly at the onset.

3.0 Mechanism and Policy

Ideally, a software system should provide security mechanisms that allow for many different (and changing) policies on security to be implemented. In particular, the software should not dictate what ‘acceptable use’ is, but should provide means by which any reasonable definition of acceptable use can be enforced.

4.0 Access security

This is a well-understood issue in software systems and there are traditional methods for resisting attempts to gain unauthorized access into a system. Login IDs, passwords, and permissions provide a mechanism that can be used to resist unauthorized access. A firewall can be used to reduce the chance of external events adversely affecting the performance of the system. Both of these mechanisms are present in the Gemini design.

5.0 Abuse security

Security from abusive actions is a more complicated matter. Aside from safety issues, which must be addressed independently of the software, there are many ways that software can help prevent abusive actions from impacting the system.

5.1 Assumptions

We have made the following assumptions about our system:

- The authorized users on our system are not malicious. Simply put, this means that, once access to the system is granted, we expect that users will not deliberately act to abuse the system.
- Even with ‘considerate’ users, accidents may happen. There is very little difference between an accident by a safe user and a malicious act, but security policies that assume one or the other may be quite different, with profound implications on the flexibility of the system.
- Since we assume non-malicious users, our policies should start with the assumption that flexibility is important (this is likely to be very true as we first learn to use the telescopes effectively), but the mechanisms should be in place to reduce misuse should situations be discovered that require a more stringent policy. (At the current time, the only non-safety security policy we have on system abuse is to protect science data.)

5.2 Policy

The general policy initially adopted by the Gemini System consists of a few rules. Although oversimplified, these rules can be stated as:

1. The science team whose instrument is ‘in the beam’ can do anything with any system in the beam.
2. Other science teams can do anything that doesn’t interfere with (1).

3. Other users can do anything that doesn’t interfere with (1) and (2).
4. The System Operator is responsible for allocating resources (assisted by the OCS) in accordance with (1), (2), and (3).
5. The only known exceptions are (a) personnel and equipment safety issues and (b) protecting data from unauthorized access before it becomes publicly available.

5.3 Implications

There are a number of implications resulting from these policies:

- While a console (engineering screen or otherwise) may require special permission to operate (e.g. normally a console might only be available for monitoring a system and may require a password to switch to control capability), once control capability has been granted then any safe action available through that console should be functional. In general, if you have permission to use a console to control a system, you should be able to do so.
- Engineering screens and OCS applications must both be able to control a system at the same time. Early on we will be learning a lot about the system and there may well be unanticipated needs for access through an engineering screen.
- Internally, systems should avoid restricting specific actions to certain modes of operation - this often replaces a mechanism for enforcing policy with a specific policy and thus that policy becomes difficult to change. Even though a particular command or action may not make sense in some modes, it is probably better to allow it unless there is a safety reason why it should not be permitted.

5.4 Mechanisms

Initially, these policies are likely to be enforced through cooperation (“Doctor, it hurts when I do this!” - “Then don’t do that...”). This will likely last up to the first time a science program is ruined because of some violation of the above rules! Consequently, while it may not be necessary to have mechanisms in operation to enforce these policies, it is important to be able to provide them later. In particular, parts of the system that are in the beam should be capable of ignoring commands from instruments that have not been allocated the beam, but this capability does not need to be active yet.

It is my belief that the EPICS Channel Access security mechanism is (assuming they fix the bugs!) sufficient to prevent most violations of the above policy, but that there is no need to enforce the policy with CA security at this time. We should know how we would use it, however. (Groups developing on non-EPICS systems may need to spend some time discussing how they might enforce the above policy should the need arise.)

Another mechanism that is available for implementing security policies is through judicious choices in what to put on a system console. If there isn’t a button on the screen, then it is extremely difficult to accidently push it.