1783 Milestone ExptsSec: S4.d

University of Alabama, Tuscaloosa, AL 35487
Xiaoyan Hong, Fei Hu, Yang Xiao, Bo Fu, Zhifeng Xiao, Songqing Yue
Sept 23, 2012

This document describes our work for milestone ExptsSec: S4.d. Work presented includes suggestions on simplifying key management based on issues identified in ExptsSec: S4.c, and further security analysis on access control.

1. Suggestions on Simplifying Key Management

In our previous report, analysis on the key management and usage are presented (see report per 1783 Milestone ExptsSec: S4.c). Issues are identified in two areas, namely, the management of multiple SSH keys and the SSH tools for remote login. Per these areas, we give suggestions to improve user experiences.

1.1 Investigation on the Management of Multiple SSH Keys

In *1783 S4.c report*, we found that "the current management of multiple private keys can result in the inconstancy of the keys distributed at different nodes, also, the potentially unnecessarily holding of outdated keys".

Our suggestions based on the current implementation of key management in ProtoGENI are given below. The first suggestion is: When a new SSH key pair is generated, to the resources that were allocated before this key generation, the newly created public key will be uploaded. Thus, the corresponding paired new private key can be used to ssh to all resources owned by the user.

The second suggestion is: For a user account, only the latest generated key pair is kept valid. Whenever there are new resources allocated, the single valid public key is uploaded to all allocated resources and users can only use the latest private key to ssh to all allocated resources. A timestamp and a "time out" period for each key could be added as an addition in dealing with this key update issue. Its usage can be separate from the expiration of the certificate that the user has.

Another way can be for the cleaning house to recognize a key being contaminated and to remove it completely.

1.2 Investigation on Remote Login Tools to GENI Nodes

In the *1783 S4.c report*, we found that "it could be uneasy for users who login to GENI resources across multiple GENI interfaces (such as Flack, ProtoGeni and Omni) and use built-in SSH tools in various operating systems. In addition, for FLACK, the browsers (tested Explore and Firefox)

were not the cause of the not-working "Visit button" and "SSH button", but operating systems were."

Suggestions: In order to simplify the use of GENI, we highly recommend to develop an easy-to-use ssh tool, and to release it with ProtoGeni or Omni, especially for users of Windows and other systems other than Linux and Mac. As to Flack, it will be desirable that the "Visit button" and the "SSH button" can work well in all operating systems.

2. GENI Authentication

We studied and investigated the access contorl mechanisms for Global Environment Network Innovation (GENI).

We studied and investigated the access control mechanisms employed by GENI projects. For access control, RBAC is the current mainstream. ABAC, however, is considered to be the future authorization policy due to its fine-grained control and high flexibility. We provide a few suggestions as follows:

Human-Robot distinguisher: Each GENI project offers a web portal through which the basic experiment management can be done. Some operations, such as editing personal/experiment information, are harmless and inoffensive. Other operations, however, could be offensive to the system. For example, a malicious experimenter may employ a robot to continuously request a resource from the aggregate manager until all of the resources are used up. These legal requests are sent through the web portal; so, the aggregate manager will not detect an anomaly. The consequence is that the malicious user keeps occupying the resource, which will never really be released. In order to distinguish a human from a robot, it is essential to perform a challenge-response test on the user for each suspicious operation. For example, if a user makes a request for resources immediately after the last owned resource is reclaimed, then a test can be conducted. One of the well-known human-robot distinguishers is CAPTCHA , which intends to generate challenges that 1) are easy enough for a human to solve and 2) can prevent software robots from filling out an online form. With the combination of a human-robot distinguisher and a limitation policy of resource usage, GENI is able to prevent resource abuse.