The Functionality-Based Application Confinement Model and its Linux Prototype FBAC-LSM

Z. Cliffe Schreuders
c.schreuders@murdoch.edu.au

Traditionally access control models have enforced restrictions primarily based on user-identities. However, user-oriented access control schemes often fail to protect users as processes do not always act on their behalf. Malicious code and software vulnerabilities both exploit a legitimate user's identity to execute malicious code. Based primarily on the identity of the executing program, application-oriented access controls restrict the damage a program can do by limiting the resources available to an individual process. Current application-oriented schemes construct policy in terms of low-level privilege requirements, therefore most users do not posses the expertise to construct and verify security policies.

I will present a new application-oriented access control model, Functionality-Based Application Confinement (FBAC), and a Linux prototype implementation, FBAC-LSM. FBAC constructs policies using policy abstractions known as *functionalities* which represent the behaviour a program is authorised to carryout. Functionalities can be parameterised with application-specific information, adjusting them to the needs of particular applications. Policies are hierarchical where high-level functionalities are composed of lower-level functionalities. For example, the Web_Browser functionally contains the lower-level http_client functionality. As policy is natively hierarchical, parts of policy for a process can be deactivated or reactivated to further restrict the resources available to applications while they are running. Uniquely, processes can have multiple restrictions which are able to simultaneously enforce multiple mandatory and discretionary controls. In this way, security goals of administrators and users can be concurrently enforced by FBAC. FBAC-LSM has been developed for Linux and will be released open source in 2009.