

Principles of Information Security CS 5323 Lecture 19

Prof. William Winsborough
November 13, 2007

Business

- For Thursday 11/15, please read
 - The Flask Security Architecture: System Support for Diverse Security Policies
 - <http://www.nsa.gov/selinux/papers/flask-abs.cfm>

13 November 2007

Winsborough CS 5323 Lecture 19

2

Practical Domain and Type Enforcement for UNIX

- DTE is a mandatory access control (MAC) mechanism that enforces policies that are quite different from Bell-LaPadula
- Good for:
 - Confining applications
 - Restricting information flow
- Based on earlier design: *simple type enforcement*
- Enhancements are designed to:
 - Simplify the use of the mechanism
 - Provide compatibility with other systems

13 November 2007

Winsborough CS 5323 Lecture 19

3

Context of Contribution

- There was recognition that various forms of MAC were needed
- Very few were commercially available
 - High costs
 - Complex administration
 - Application incompatibility
 - User training
- Can security enhancement be added to mainstream OS's in a way that is
 - Understandable?
 - Effective?
 - Unobtrusive?

13 November 2007

Winsborough CS 5323 Lecture 19

4

DTE Enhancements to Simple Type Enforcement

- Policies are expressed in high-level language DTEL
 - Allows DTE to be superimposed on applications that are not aware of DTE
- During system execution, configuration information is maintained in a small policy database
 - Much configuration information is given implicitly by the location of files in the directory hierarchy
 - Facilitates compatibility with systems that are not DTE-aware

13 November 2007

Winsborough CS 5323 Lecture 19

5

Type Enforcement Background

- Invariant AC attributes
 - A *domain* is associated with each subject
 - Permissions are granted to domains
 - A *type* is associated with each object
 - Permissions are granted on types
- A Domain Definition Table (DDT) specifies access modes available based on domain and type
- Subject-to-Subject access control is based on a Domain Interaction Table (DIT)
 - Example access modes: Signal, create, destroy

13 November 2007

Winsborough CS 5323 Lecture 19

6

Issues with Simple Type Enforcement (TE)

- Configuration may become too complex
 - Tables become huge
- Table structure does not map naturally to standard system structures
 - Object (file) hierarchy
 - Subject (process) hierarchy
 - System structures are relevant to security
- While DoD has a tradition of defining BLP and DAC policies, no tradition exists for TE

13 November 2007

Winsborough CS 5323 Lecture 19

7

DTE Difference

- High-level language expresses reusable AC configurations
- Many type/file associations are maintained implicitly
 - Simplifies specification
 - Don't have to store label with file
 - File format does not change (remains UNIX)
 - Permits application of DTE policies to existing media

13 November 2007

Winsborough CS 5323 Lecture 19

8

DTE Language (DETL)

- Human friendly (ascii)
- Replaces DDT and DIT tables
- Statements:
 - Type: declares types
 - Domain: declares domains and defines
 - Entry points
 - Object access rights (modes x types)
 - Subject access rights (exec vs. auto modes)
 - Initial_domain: domain of initial process
 - Assign: Associates locations in object hierarchy with specific types
 - -r associates entire subtree rooted at given point
 - -s cannot create objects of other types in that area

13 November 2007

Winsborough CS 5323 Lecture 19

9

Controlled Invocation

- Compared with regular UNIX
 - UNIX setuid allows certain entry points to be run with high privilege
 - DTE exec allows certain entry points to be run in certain domains

13 November 2007

Winsborough CS 5323 Lecture 19

10

Policy Maintenance

- Type labels are not stored with file's on-disk representation
 - Expensive to reconfigure
 - Changes low-level formats, creating incompatibility with non-DTE systems (like distributed file systems)
- UNIX kernel-resident runtime policy database
 - Established at boot time
 - Small (because of implicit associations)
 - Fits in memory—no extra I/O

13 November 2007

Winsborough CS 5323 Lecture 19

11

Policy Maintenance

- Changes to policy db:
 - File creation
 - Add assign stmt if implicit type in new location is not correct
 - File renaming
 - Type should not change
 - May have to add assign stmt if implicit type in new location is not correct
- Recoverability
 - Log file + Snap shot = reboot policy db
- Efficiency
 - Minimal additional I/O for security
 - No addition I/O when normal operation does not do I/O

13 November 2007

Winsborough CS 5323 Lecture 19

12