TRUSTED
COMPUTER SOLUTIONS

Trusted Computer Solutions, Inc.
2350 Corporate Park Drive, Suite 500
Herndon, VA 20171 USA

V: 703.318.7134
F: 703.318.5041
www.TrustedCS.com

SELinux and MLS

Chad Hanson
Trusted Computer Solutions, Inc.
121 W Goose Alley
Urbana, IL 61801 USA
chanson@TrustedCS.com

Today's Presentation

- What is MLS?
- MLS and the SELinux Security Context
- MLS Privileges
- Defining the MLS Security Policy
- MLS Users
- MLS Security Class
- Current MLS Status
- Certification Goals

What is MLS?

- Multi-Level Security (MLS) Theory
  - Bell-LaPadula Model (BLP)
    - Confidentiality
    - Policy Rules (No Read Up, No Write Down)
      - Simple Security Property (No Read Up)
      - *-Property “Star-Property” (No Write Down)
    - Additionally restrict write to equality

- MLS is a complimentary module to Type Enforcement (TE)
  - Can describe complex confidentiality relationships
  - Utilized by the DoD and Intelligence Community

MLS and SELinux Security Context

- MLS is added to end of security context
- Subject has MLS Range
  - Effective (low)
  - Clearance (high)

Example:

```
system_u:system_r:initrc_t:s0-s2
```

- Objects
  - Single Level, directories maybe allowed a range
  - Objects can specified as “trusted” to allow subject access

Example:

```
system_u:object_r:initrc_exec_t:s0
```

MLS Privileges

- MLS Privilege Model
  - A fine-grained set of MLS privileges is implemented using type attributes

    Examples:

    ```
    attribute mlsfileread;
    attribute mlsfilewrite;
    ```

  - Processes gain the use of an MLS privilege by executing in a domain which has the associated attribute.

    Example:

    ```
    typeattribute my_priv_domain_t mlsfileread;
    ```

  - The MLS security policy defines the affects of these privilege attributes on the outcome of security decisions.

Defining the MLS Security Policy

- MLS Security Policy is defined through an extension of the SELinux constraint language
- MLS constraint rules define requirements based on a class – permission pairing
- Highly flexible to meet custom policy needs
- New process expressions:
  - L1, H1 & L2, H2– MLS range (low, high) of Context 1 & 2
- Label Comparison Operators : eq, dom, domby, incomp

Example:

```
mlsconstrain file read
    (( l1 dom l2) or
        (( t1 == mlsfilereadtoclr ) and ( h1 dom l2 )) or
        ( t1 == mlsfileread ));
```

Defining the MLS Security Policy (cont'd)

- MLS transition validation rules define additional requirements for upgrading or downgrading an object
- Not tied to a particular permission; they are defined for the security class as a whole
- Triplet (object 1, object 2, process)
- New Process Expressions
  - U3, R3, T3 – User, Role and Type of Context 3

Example:

```
mlsvalidatetrans file
  (( l1 eq l2) or
   (( t3 == mlsfileupgrade ) and ( l1 domby l2 )) or
   ( t3 == mlsfiledowngrade ));
```

MLS Users

- User Database
  - MLS User attributes consist of default label and label range
  - Default Label
    - Used for system login
  - Label Range
    - Minimum Label
    - Maximum Label (clearance)

Examples:

```
user jdoe { user_r } level s2 range s2;
user jadmin{ user_r secadmin_r } level s2 range s0 -
  s9:c0 . c127;
```

MLS Security Class

- MLS Security Class
  - A set of permissions for MLS actions performed by applications
    - Bind_all_levels
      - Application can bind to all port labels
    - Translate_all_levels
      - Application can translate all levels

  - Utilized by Security Administrator Role and others
    - Applications such as “ls”
      - Allow translation of object labels which are not dominated by current process

      Example:

      `an UNCLASSFIED process seeing a CONFIDENTIAL label`

Current MLS Status

- Most changes in SELinux CVS tree on SourceForge
  - MLS policy language
  - MLS enhancements to policy and user utilities
  - Runtime detection and enforcement of MLS based on policy

- Upcoming Contributions
  - range_transition rule
  - Ported XACE to X.org 6.8.2 release
    - Working with community for acceptance
  - MLS security class

- Future Planned Contributions
  - File and Network Polyinstantiation
  - Network Flow Control

Certification Goals

- Driven by customer requirements
- Common Criteria
  - Labeled Security Protection Profile (LSPP)
  - Controlled Access Protection Profile (CAPP)
  - Role-Based Access Control Protection Profile (RBAC)
  - Assurance Level of EAL 4
- DCID 6.3 PL 4

Thank You

Simplify Secure Information Sharing

www.TrustedCS.com