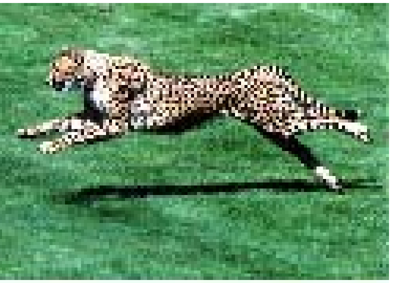


Label-Based Access Control With IDS Cheetah

Jonathan Leffler

IBM Informix Dynamic Server

Agenda

- What is Label-Based Access Control?
- Why would you use LBAC?
- Mechanics of LBAC
  - Creating a Security Policy
  - Applying the Policy
  - Granting Labels
  - Granting Exemptions
  - DML operations
- The DBSECADM role
- Backups, Restores, Imports and Exports
- Questions and Answers

What is Label-Based Access Control?

- LBAC in IDS is based on the design in DB2
  - Essentially the same
- Provides Mandatory Access Control (MAC)
  - Orange Book (B1) style label-based security
    - But different from MLS – multi-level security – in some aspects
  - Analogous to Oracle’s Label Security
  - Intended for certification versus Common Criteria
    - At EAL4, against LSPP (Labelled-Security Protection Profile)
- Based on the Bell-LaPadula model
  - Bell, LaPadula, ‘Secure Computer System: Unified Exposition and Multics Interpretation’, ESD-TR-75-306, Mitre Corporation, 1976.

What is Label-Based Access Control?

- Data is labelled
  - By columns,
  - Or by rows
  - Or both
- Users are granted labels
- Access to data is controlled
  - Based on user’s label
  - And data’s label
  - Read permissions different from write permissions
    - If required

LBAC Demonstration

- User Label – Public

Security Label	Name	Rank	Task
Public	John Smith	CEO	Run Company
Public	James Talbot	CFO	Run Accounts
Public	Malcolm Knight	CIO	Run IT

- SELECT * FROM People

LBAC Demonstration

- User Label – Confidential

Security Label	Name	Rank	Task
Public	John Smith	CEO	Run Company
Confidential	Heinrich Messier	Accountant	SEC Relations
Public	James Talbot	CFO	Run Accounts
Confidential	Jessica McHenry	IT Specialist	Networks
Confidential	Melissa Williams	IT Specialist	Databases
Public	Malcolm Knight	CIO	Run IT

- SELECT * FROM People

LBAC Demonstration

- User Label – Secret

Security Label	Name	Rank	Task
Secret	Verity Dolittle	M & A	Buy Google
Public	John Smith	CEO	Run Company
Confidential	Heinrich Messier	Accountant	SEC Relations
Secret	Alex Grimwald	M & A	Buy Yahoo!
Public	James Talbot	CFO	Run Accounts
Confidential	Jessica McHenry	IT Specialist	Networks
Secret	Jane Ferguson	M & A	Buy Microsoft
Confidential	Melissa Williams	IT Specialist	Databases
Secret	Kate Ball	M & A	Buy Oracle
Public	Malcolm Knight	CIO	Run IT

- SELECT * FROM People

Agenda

- What is Label-Based Access Control?
- Why would you use LBAC?
- Mechanics of LBAC
  - Creating a Security Policy
  - Applying the Policy
  - Granting Labels
  - Granting Exemptions
  - DML operations
- The DBSECADM role
- Backups, Restores, Imports and Exports
- Questions and Answers

Why would you use LBAC?

- LBAC provides extra control over who can see what
  - Labels can be applied to columns
    - To prevent unauthorized users from reading those columns.
  - Labels can be applied to rows
    - Unauthorized users will not see rows with too sensitive labels
- Access can be controlled declaratively
  - By granting or revoking labels
  - By granting or revoking exemptions
- Provides extra protection for the most sensitive data
  - Credit card numbers
  - Social security numbers

Why would you use LBAC?

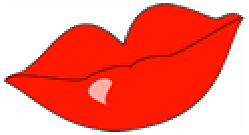
- LBAC controls prevent unauthorized access
  - By system administrators
  - As well as by users

Agenda

- What is Label-Based Access Control?
- Why would you use LBAC?
- Mechanics of LBAC
  - Creating a Security Policy
  - Applying the Policy
  - Granting Labels
  - Granting Exemptions
  - DML operations
- The DBSECADM role
- Backups, Restores, Imports and Exports
- Questions and Answers

Mechanics of LBAC

- LBAC can be frightening
  - Don’t be scared
  - Don’t use it if you don’t need it
- LBAC has
  - Security policies
  - Security label components
  - Security labels
- Designing policies
  - Work out the types of protection needed
    - Create the security label components
    - Create the security policy
    - Create the security labels
  - Remember the KISS Principle
    - Keep It Simple, Stupid!



Creating a Security Policy

- Security Policy
  - Database object that can be applied to tables
  - Composed of one or more security label components
  - Created after label components
- Security Label Component
  - Components can be Array, Set, or Tree types
  - Controls which rules apply to access

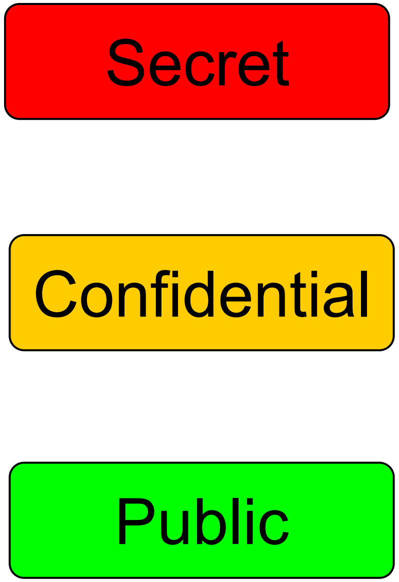
Creating a Security Policy

- Security Label
  - Always associated with a specific security policy
  - Includes one value for each component in the policy
    - A value is zero or more of the elements of the component
  - Created after security policy
  - Labels apply to users
    - Subject labels
  - Labels apply to data
    - Object labels

Array Component

- Ordered list of elements
  - Up to 64 elements
  - First one is the highest
- Only one element allowed in a label for a component
- Read data that is less than or equal to your level
  - No read up
- Write data equal to your level.
-
  ```
  CREATE SECURITY LABEL COMPONENT level
  ARRAY ['Secret', 'Confidential',
  'Public'];
  ```

Security Label Component – Array



- If your read label is Secret
  - You can read anything
  - Your write label must be Secret
- If your read label is Confidential
  - You can read Confidential or Public data
  - Your write label must be Confidential
- If your read label is Public
  - You can only read Public
  - Your write label must be Public
- Closest to the Bell-LaPadula model

Set Component

- Non-ordered set of elements
  - Up to 64 elements
- One or more elements in a label for a component
- You can read or write data if your label contains **all** the elements in the data label
-
```
CREATE SECURITY LABEL COMPONENT
department SET {'Marketing', 'Product
Development', 'Quality Assurance'};
```

Security Label Component – Set

Marketing

Product Development

Quality Assurance

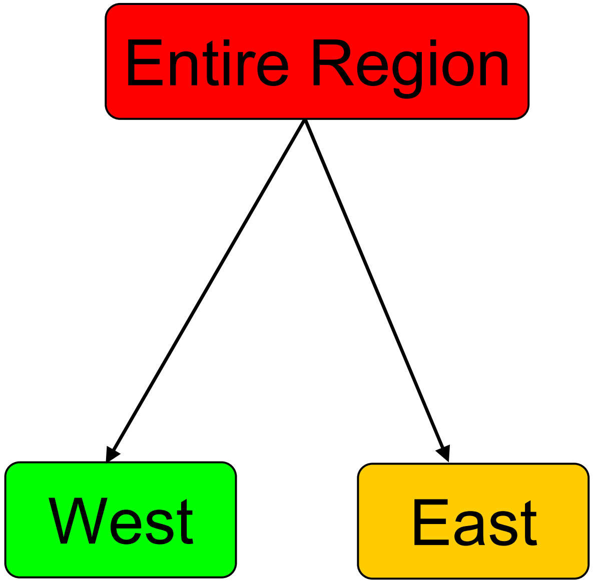
- If your read label has { Marketing }
  - You can read anything that is marked just Marketing
  - Or has an empty Department label component
  - But not anything marked Marketing and Product Development
- If your read label is { Product Development, Quality Assurance }
  - You can read items marked Product Development
  - You can read items marked Quality Assurance
  - Or both
  - Or with an empty label component

Tree Component

- Hierarchical set of elements
  - Up to 64 elements
- You can have one or more elements in a label
- You can read or write data if your label contains **any** of the elements in the data label, or the ancestor of one such element.

-
  ```
  CREATE SECURITY LABEL COMPONENT region
  TREE (‘Entire Region‘ ROOT, ‘East’
  UNDER ‘Entire Region’, ‘West’ UNDER
  ‘Entire Region’);
  ```

Security Label Component – Tree



- If your read label is Entire Region
  - You can read anything
- If your read label is West
  - You can only read West or empty
- If your read label is West, East
  - You can only read East or West
  - You cannot read items labelled Entire Region

Read Access Rules

- Read Access Rules are applied when data is read.
  - Data is read on SELECT, UPDATE and DELETE.
  - **IDSLBACREADARRAY**
    - Each array component of the user security label must be greater than or equal to the array component of the data security label
    - The user can only read data at or below his/her level.
  - **IDSLBACREADSET**
    - Each set component of the user security label must include the all the set components of the data security label.
  - **IDSLBACREADTREE**
    - Each tree component of the user security label must include at least one of the elements in the tree component of the data security label (or the ancestor of one such element).

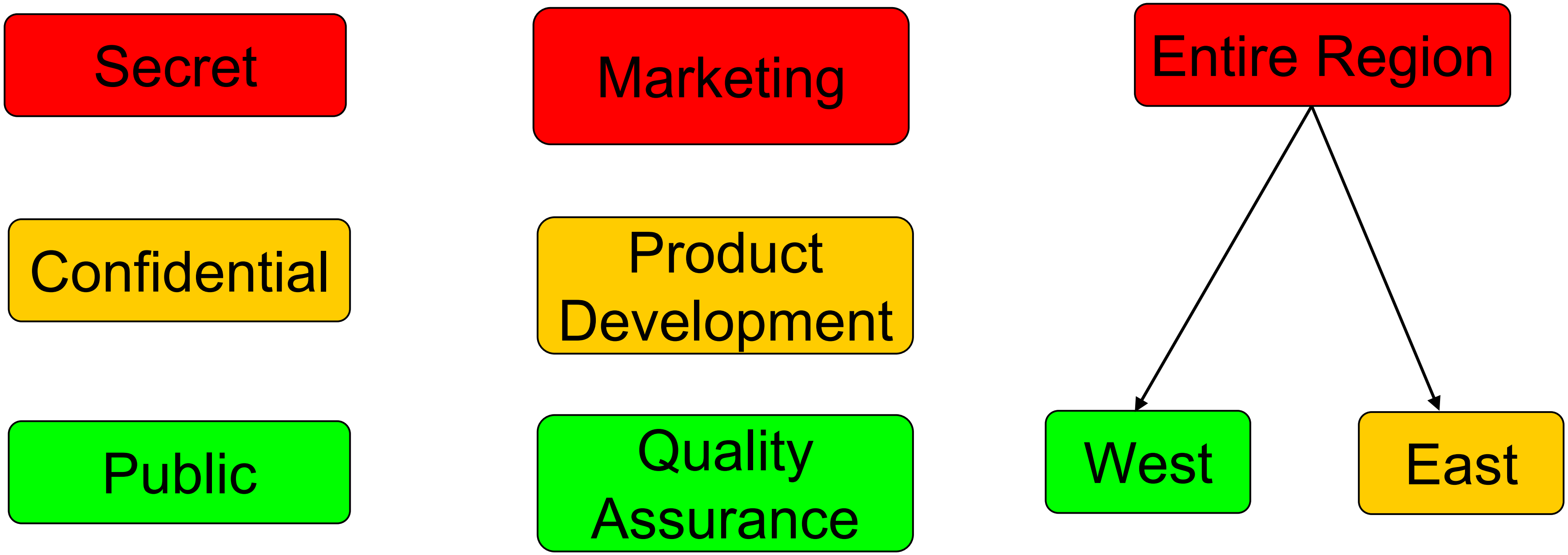
Write Access Rules

- Write access rules are applied when data is written.
  - Data is written on INSERT, UPDATE and DELETE.
  - **IDSLBACWRITEARRAY**
    - Each array component of the user security label must be equal to the array component of the data security label
    - That is, the user can write data only at his/her level.
    - No write-down.
  - **IDSLBACWRITESET**
    - Each set component of the user security label must include the set component of the data security label.
  - **IDSLBACWRITETREE**
    - Each tree component of the user security label must include at least one of the elements in the tree component of the data security label (or the ancestor of one such element).

Creating a Security Policy

- A Security Policy is created from Security Label Components
  - Up to 16 components
  - But think in terms of using 1 to at most 3
- **`CREATE SECURITY POLICY company COMPONENTS level, department, region;`**
  - This policy has three components.
  - Labels for this policy have a value for each component
    - Zero or more elements for the tree and set components
      - Department
      - Region
    - Precisely one element for the array component
      - Level

Multi-Component Security Policy



Secret

Confidential

Public

Marketing

Product Development

Quality Assurance

Entire Region

West

East

- Each label will have one of Secret, Confidential, Public
- Each label will have zero or more of Marketing, Product Development, Quality Assurance
- Each label will have zero of mode of Entire Region, East, West

Creating Security Labels

- A Security Label specifies the value for each component in a Security Policy

-
  ```
  CREATE SECURITY LABEL company.director
      COMPONENT level 'Secret’,
      COMPONENT department 'Product
        Development', 'Quality Assurance',
      COMPONENT region 'Entire Region';
  ```
  - Single element in label for components level and region
  - Multiple elements in label for component department

Applying a Security Policy

- Protecting rows
  - Row level protection granularity
  - Attach security policy to table
  - Add security label column of new type:
    - IDSSECURITYLABEL
- Protecting columns
  - Column level protection granularity
  - Attach security policy to table
  - Attach security label to one or more columns
- Protecting rows and columns
  - Attach security policy to table
  - Apply label to one or more columns
  - Add security label column

Protecting a Table

-
  ```
  CREATE TABLE T1
  ( C1 IDSSECURITYLABEL, { Always NOT NULL }
    C2 INTEGER NOT NULL,
    C3 CHAR(10) NOT NULL
       COLUMN SECURED WITH director
  ) SECURITY POLICY company;
  ```

-
  ```
  ALTER TABLE T2
    ADD(C1 IDSSECURITYLABEL),
    MODIFY(C2 INTEGER NOT NULL
              COLUMN SECURED WITH manager),
    ADD SECURITY POLICY company;
  ```

Granting Security Labels

- Security labels can be GRANTED to users.
  - The same label can be granted to many users.
  - Each user can be granted 0, 1 or 2 labels for a policy
    - One READ label
    - One WRITE label
  - If the labels are different,
    - The READ label must dominate the WRITE label.
  - Labels may not be granted to roles.
- **GRANT SECURITY LABEL company.director TO mr_ceo FOR ALL ACCESS;**
  - Cannot read Marketing information

Granting Exemptions

- The read-write rules are quite stringent
  - No read up (within an array or tree)
  - No write down (within an array or tree)
- Sometimes, data needs to be reclassified.
- EXEMPTIONS allow you to control that.
  - A user can be granted an exemption to bypass one or more access rules for a component type in a security policy
- Only someone with DBSECADM authority can grant exemptions
  - DBSECADM cannot grant themselves labels or exemptions

Granting Exemptions

-
  ```
  GRANT EXEMPTION ON RULE
          IDSLBACWRITEARRAY WRITEDOWN
          FOR company TO mr_ceo;
  ```
  - User ‘mr_ceo’ can now breach the array rule
  - He can write a row with level ‘public’ or ‘confidential’.
  - Must still keep to the rules for department and region
- Exemptions should only be granted temporarily
  - Otherwise, why bother with LBAC at all!

-
  ```
  REVOKE EXEMPTION ON RULE
          IDSLBACWRITEARRAY WRITEDOWN
          FOR company FROM mr_ceo;
  ```

SQL Function: SECLABEL_BY_COMP

- **SECLABEL_BY_COMP**
  - Security label value based on policy and components
  - A built-in function
  - Used in insert and update operations
  - Returns the row security label of a data row
    - Specified by its individual components
  - INSERT INTO T1 VALUES (SECLABEL_BY_COMP(‘company', ‘Director:Marketing:West’), 1, 'xyz')

SQL Functions

- **SECLABEL_BY_NAME**
  - Security label value based on policy and label name
  - A built-in function
  - Used in insert and update operations
  - Returns the row security label of a data row
    - Specified by label name
  - UPDATE T1 SET C1 = SECLABEL_BY_NAME(‘company', ‘manager')

SQL Functions

- **SECLABEL_TO_CHAR**
  - Convert security label value to component string
  - A built-in function used in select operations
  - To retrieve the row security label column
  - And returns it as a string
  - SELECT SECLABEL_TO_CHAR(‘company', C1), C2, C3 FROM T1

Agenda

- What is Label-Based Access Control?
- Why would you use LBAC?
- Mechanics of LBAC
  - Creating a Security Policy
  - Applying the Policy
  - Granting Labels
  - Granting Exemptions
  - DML operations
- The DBSECADM role
- Backups, Restores, Imports and Exports
- Questions and Answers

DBSECADM

- Database security administrator
- Server level role
- Can be granted by DBSA only
- Responsibilities
  - Create, drop, alter and rename security label components
  - Create, drop and rename security policies
  - Create, drop and rename security labels
  - Attach, detach policies to/from tables
  - Grant and revoke security labels
  - Grant and revoke policy exemptions
  - Grant and revoke setsessionauth privilege

SETESSIONAUTH privilege

- SET SESSION AUTHORIZATION statement
  - Allows DBA to assume identity of another user
    - Without any authentication step
  - ∴ DBA can see other users' data in protected table
- New privilege SETSESSIONAUTH
  - To prevent unauthorized access
  - Allows DBSECADM to control who can use
    - SET SESSION AUTHORIZATION
  - Not automatically granted to DBA

SETESSIONAUTH privilege

- Only DBSECADM can grant SETSESSIONAUTH
- Only users with SETSESSIONAUTH privilege
  - Can use the SET SESSION STATEMENT
- During conversion from older server to 11.10
  - SETSESSIONAUTH privilege is granted to DBA
  - For backward compatibility.

Agenda

- What is Label-Based Access Control?
- Why would you use LBAC?
- Mechanics of LBAC
  - Creating a Security Policy
  - Applying the Policy
  - Granting Labels
  - Granting Exemptions
  - DML operations
- The DBSECADM role
- Backups, Restores, Imports and Exports
- Questions and Answers

Utilities

- dbschema, dbexport, dbimport
  - User must be granted DBSECADM role if database contains LBAC objects
  - User must have necessary labels or exemptions if all rows in protected tables are to be exported/imported
- onload, onunload
  - User must possess all exemptions to bypass the security policy

Utilities

- HPL (Express mode)
  - User must possess all exemptions to bypass the security policy
- All other load/unload utilities
  - User must have necessary labels or exemptions if all rows in protected tables are to be loaded/unloaded

Restrictions

- Tables that cannot be protected
  - VTI tables
  - Tables with VII indexes
  - Temp tables
  - Typed tables
  - Hierarchical tables
- Security label column cannot have
  - Referential constraints
  - Check constraints
  - Primary Key or Unique constraints
    - If security label column is the only column in constraint
  - Column protection
  - Encryption

Is that all?

- No.
- There are lots of other operations
  - ALTER, RENAME, DROP operations
- There are other issues to discuss
  - ER, HDR
- Performance issues
  - SELECT COUNT(*) FROM RowProtectedTable
    - Slower than on non-protected tables

http://www.ibm.com/software/data/informix

http://www.ibm.com/software/data/informix

Thanks to Dave Desautels and Lynette Adayilamuriyil for the draft presentation