DTEvisual User Guide

Contents

List of Figures

1 Overview

1.1 Introduction

DTEvisual is a user-level policy development system. The purpose of DTEvisual is to help students better understand the Domain and Type Enforcement (DTE) access control model. DTE is well suited to protecting system level processes and configuration information. Writing a complex DTE specification may be a daunting task for beginners; however, the visualization and interaction features of this system will facilitate the learning process.

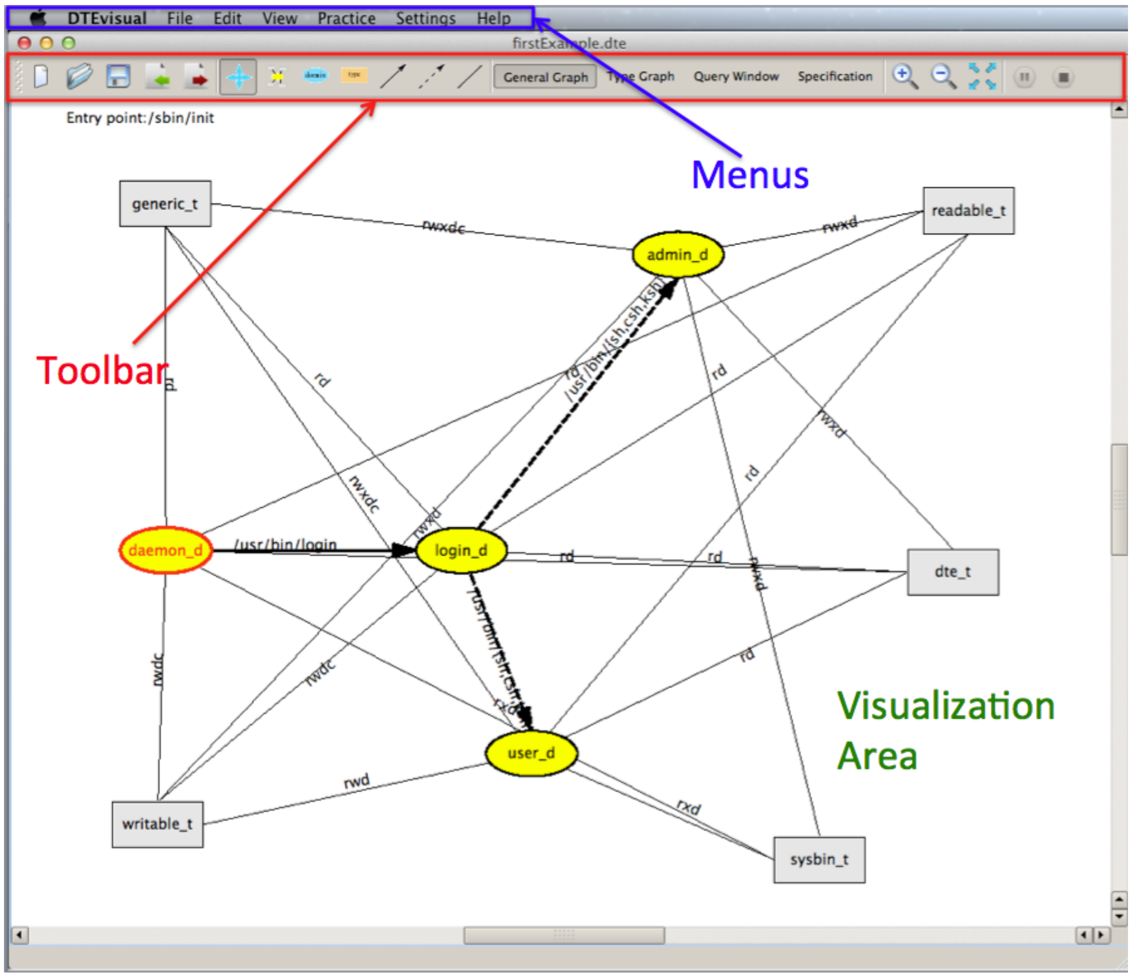


Figure 1: Main User Interface

1.2 Features

Following are the main features of DTEvisual:

1) Users can design and edit a specification visually and then save the resulting policy.
2) DTEvisual allows users to display a specification using two types of graphs: the General Graph and the Type Graph.

3) Users can use DTEvisual to carry out queries about a specific policy (e.g., what is the type of some specific file).

1.3 User Interface

DTEvisual provides separate File, Edit, View, Practice, Settings and Help menus, as well as a toolbar and Shortcuts. The File, Edit and View menus are used to access functionality related to viewing and editing specifications. The Settings menu is available for enabling the Query Animation function and setting animation intervals. The Practice menu allows students to access a test for evaluating student learning.

Figure 1 shows the main user interface of DTEvisual, which contains the menus, toolbar, and visualization area. The toolbar is shown in more detail in Figure 2.

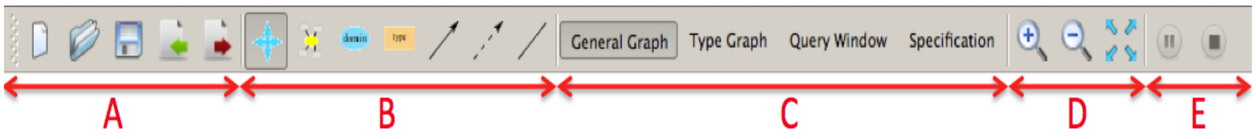


Figure 2: Toolbar

Commonly used operations are classified into the following five groups in the toolbar.

1) Group A has I/O-related icons (e.g., importing and exporting a file).
2) Group B has operators for changing modes and editing (e.g., moving nodes, highlighting, and adding domain and type nodes).
3) Group C contains shortcuts to different functions for policy analysis (e.g., opening the query window).
4) Group D contains icons for the user to zoom in and out and enter/exit the full screen mode.
5) Group E has two icons for controlling query animation.

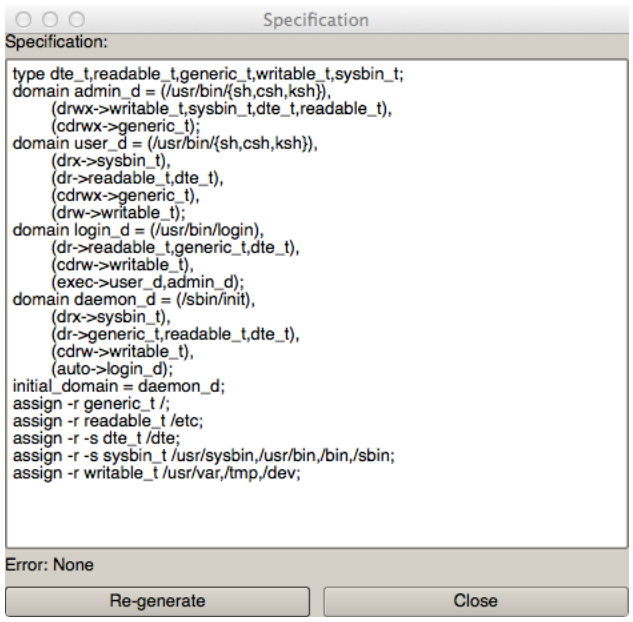
2 File Types

DTEvisual supports two types of files: DTE specifications in *.dte files and visualization descriptions in *.dtevis files. A dte file contains a DTE specification in text format. A dtevis file records a DTE specification as well as the layout of the General Graph and Type Graph.

The user may import a DTE specification in either format, visually modify it, and save[1] the result to a dtevis file or export it to a dte file. DTEvisual also allows the user to create a DTE specification graphically, and save the design to a dtevis file or export it to a dte file.

[1]The dtevis file is considered to be the native format. A dtevis file is opened or saved and one imports or exports a dte file.

3 Specification



```
type dte_t,readable_t,generic_t,writable_t,sysbin_t;
domain admin_d = (/usr/bin/{sh,csh,ksh}),
        (drwx->writable_t,sysbin_t,dte_t,readable_t),
        (cdrwx->generic_t);
domain user_d = (/usr/bin/{sh,csh,ksh}),
        (drx->sysbin_t),
        (dr->readable_t,dte_t),
        (cdrwx->generic_t),
        (drw->writable_t);
domain login_d = (/usr/bin/login),
        (dr->readable_t,generic_t,dte_t),
        (cdrw->writable_t),
        (exec->user_d,admin_d);
domain daemon_d = (/sbin/init),
        (drx->sysbin_t),
        (dr->generic_t,readable_t,dte_t),
        (cdrw->writable_t),
        (auto->login_d);
initial_domain = daemon_d;
assign -r generic_t /;
assign -r readable_t /etc;
assign -r -s dte_t /dte;
assign -r -s sysbin_t /usr/sysbin,/usr/bin,/bin,/sbin;
assign -r writable_t /usr/var,/tmp,/dev;
```

Figure 3: Specification Window

DTEvisual allows you to inspect the text version of the DTE specification. The specification window is depicted in Figure 3. The specification window is accessible from the toolbar or the View menu (Figure 4). Figure 3 shows an example specification. The detailed syntax for the specification file can be found in Appendix A.

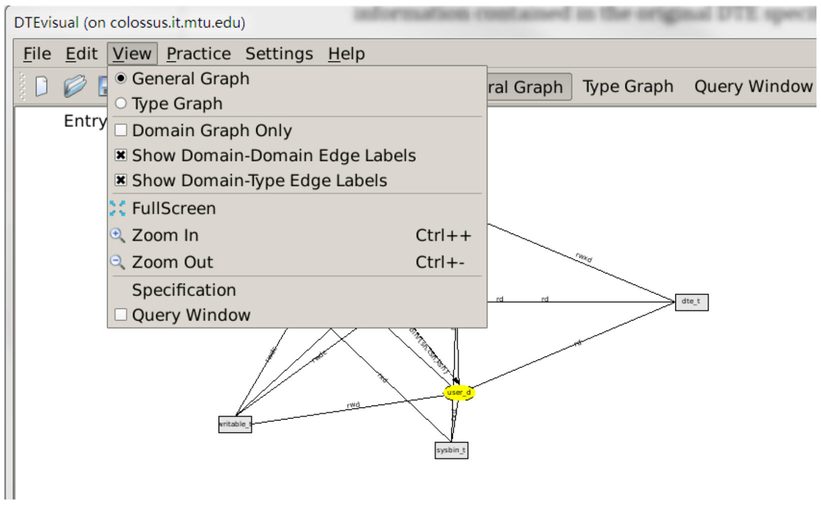


Figure 4: View Menu

4 General View Operation

View operations include Zoom In/Zoom Out either using the magnifier buttons on the toolbar or through View Menu -> Zoom In/Zoom Out. CTRL+/- are shortcuts for the zoom in and zoom out operations respectively.

5 File Operations

File operations accessible from the toolbar as well as from the File menu as indicated below:

- To start from scratch
  File Menu -> New
- To import/export a DTE specification (*.dte)
  File Menu -> Import/Export
- To open/save a DTE diagram (*.dtevis)
  File Menu -> Open/Save/Save As

When a DTE specification is imported, the system will automatically layout the General Graph so that visual clutter is minimized. In this process, a real-time animation is visible that shows how the nodes in the general graph change their positions. When the layout seems good enough, the process can be stopped by hitting the space bar. The Type Graph is constructed at the same time without user input.

6 Normal Mode

After importing an existing specification or diagram, the system comes up in Normal Mode. This mode can also be selected from the toolbar. In this mode, one can view a policy from different perspectives. Certain editing operations are also available from context menus.

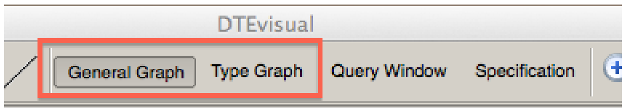


Figure 5: Graph Selection from Toolbar

6.1 General Graph and Type Graph

DTEvisual provides two visualizations of a DTE specification: the General Graph and the Type Graph. Users can switch between the two graphs by clicking buttons in the toolbar (Figure 5). These operations are also accessible from the View menu.

6.1.1 General Graph

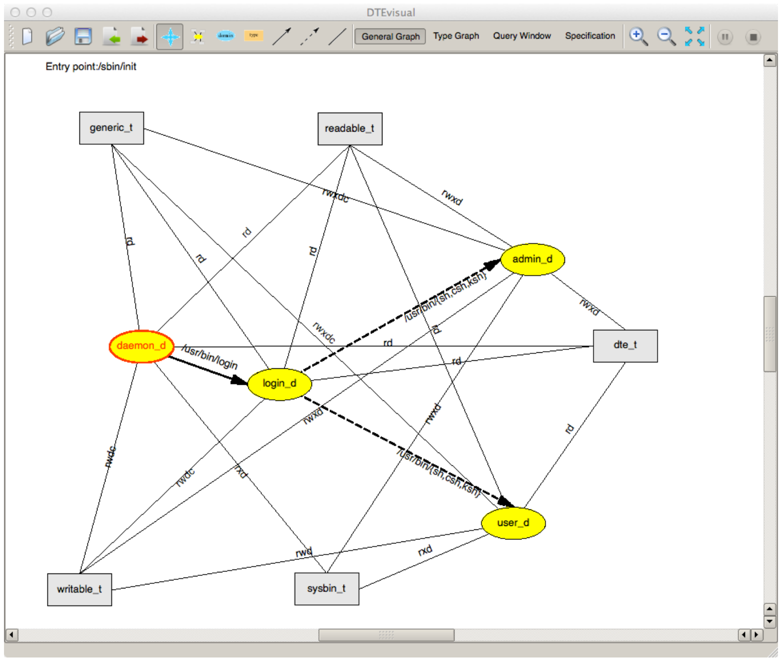


Figure 6: General Graph

The General Graph depicts domains, types, transitions between domains, and access of domains to a given type. Figure 6 is an example General Graph[2].

Each elliptical node corresponds to a domain. Each square node corresponds to a type. There is an undirected edge between domain node D and type node T if D has any access to type T. The edge is labeled to indicate the type of access that is allowed. There is a directed edge from domain node D1 to domain node D_2 if D_1 has either auto or exec access to node D_2. When the access is through the auto flag, the edge between nodes is solid. When the access is through the exec flag, the edge is dashed.

In Normal Mode, users can drag the existing nodes in the General Graph to change their positions.

[2]The specification for the depicted graph is given in Figure 3

6.1.2 Type Graph

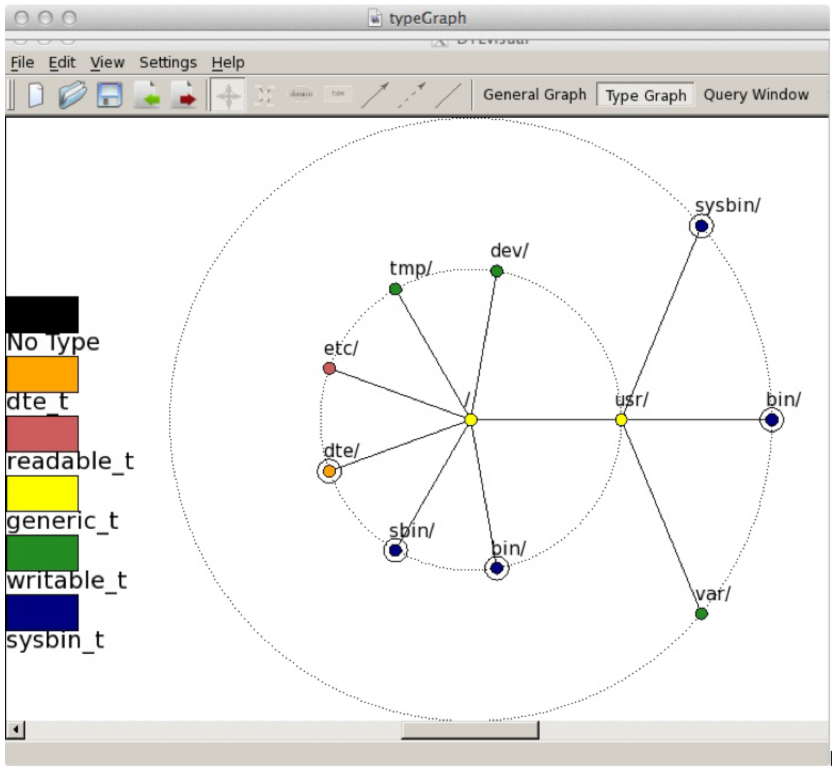


Figure 7: Type Graph

The Type Graph shows object types and is displayed using a radial tree. All the files at the same level in the directory hierarchy are connected using a dotted circle. Directories are indicated by a trailing slash. Different types are identified through color of the labels. Figure 7 is an example Type Graph[3].

When a directory node D in the graph has no children, all files beneath D have the same type as D. When a directory node D has children C_1, C_2, ..., C_N. All files beneath D except for files C_1, C_2, ..., C_N and their descendants in the directory hierarchy have the same type as D.

6.2 Visualization Customization

DTEvisual allows a user to customize the General Graph. Following are the customization functions and the View menu options by which they are accessed. These options are shown in Figure 4.

1) Domain Graph Only : shows only the domains without interconnecting edges

2) Show Domain-Domain Edge Labels: adds edges between domains corresponding to transitions allowed by the specification

3) Show Domain-Type Edge Labels: adds edges between domains and types corresponding to accesses allowed by the specification.

[3]The specification for the depicted graph is given in Figure 3

7 Highlight Mode

Since a General Graph may become cluttered when the number of nodes in it increases, DTEvisual allows users to highlight part of the graph to alleviate this problem. This functionality can be accessed through the Edit menu (Edit -> Highlight Mode) or by clicking the highlight mode button on the toolbar.

The first time a user clicks on a node/edge, the item itself and its adjacent edges/nodes are highlighted while the rest of a general graph is grayed out. The second click on the same item does the opposite: only the rest of a General Graph is highlighted. The third click brings the rendering back to Normal Mode. Figures 8 to 10 demonstrate the highlighting operation step by step as below.

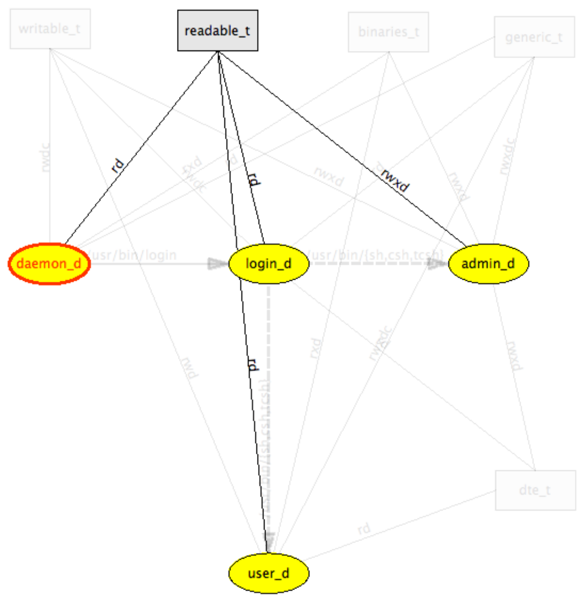


Figure 8: First Click

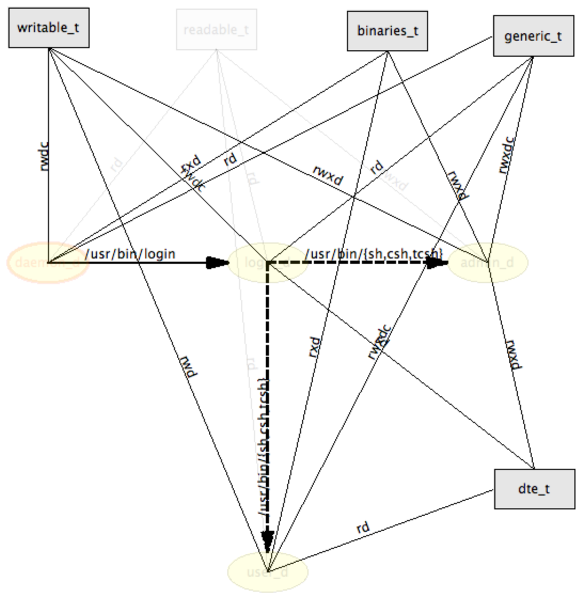


Figure 9: Second Click

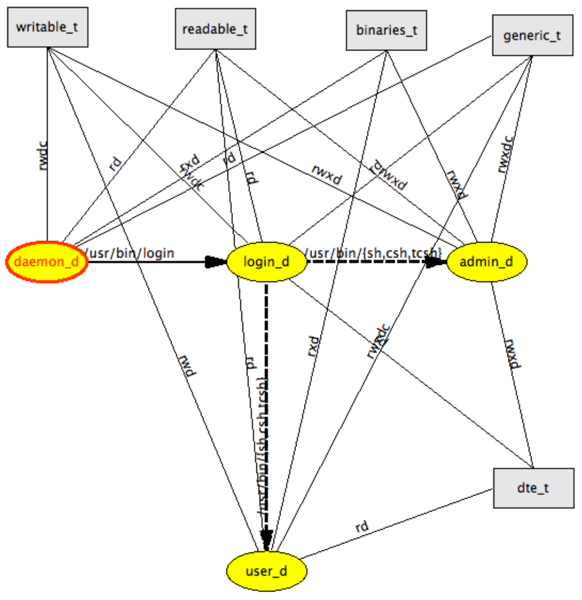


Figure 10: Third Click

8 Edit Operations

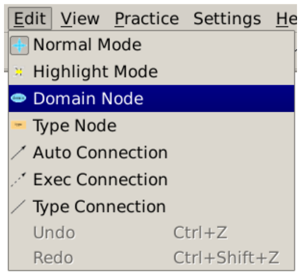


Figure 11: Edit Menu

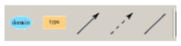


Figure 12: Edit Buttons on Toolbar

In this section, we will introduce the edit operations, which are available in the Edit Menu (Figure 11), the toolbar (Figure 12), and the Context Menu (which pops up when clicking or right clicking in within blank space or on an object in the visualization area).

The buttons on the toolbar can facilitate edit operations like drawing various lines and switching between the General Graph and Type Graph.

Please note that the Edit Menu and related operations are only available for the General Graph. Type Graph edit operations are only available from the Context Menu. Type Graph operations are described in section 8.2.

8.1 General Graph

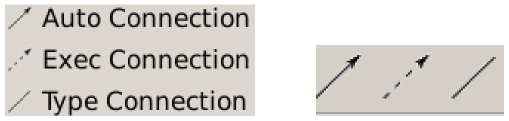
Users can add domain nodes, type nodes, or edges between them.

8.1.1 Adding Elements

A Domain Node or Type Node is added be first selecting the type of element to be added as indicated below:

- Domain Node: Use Edit -> Domain Node or click from the toolbar.
- Type Node: Use Edit -> Type Node or click from the toolbar.

The node is placed by clicking on a blank spot in the visualization area (Figure 1). A non-empty name is then required for the newly added node.



(a) Edit Menu Options (b) Toolbar Shortcuts

Figure 13: Adding Elements

To add an edge between two nodes, users must first select the correct connection type from Edit Menu or click the appropriate buttons on the toolbar, and then drag the mouse from one node to the other. (See Figure 13) recent answer being highlighted.

The following rules apply to addition of edges:

- No edge will be added if users attempt to add a wrong type of edge between the two nodes. (For example, auto connections cannot be added between a domain node and a type node.)
- For edges between domain nodes, an entry point program whose matches the format of a Linux file name must be specified.
- For edges between domain nodes and type nodes, users must specify valid permissions which should only contain the characters 'c', 'r', 'w', 'x' and 'd'.
- No two edges of the same type are allowed between any two nodes.

8.1.2 Modifying Elements

Users can also delete and change the name of a node or edge in a General Graph. This is done through the Context Menu of the node or edge. Once a node is deleted, all the edges adjacent to the node are also deleted. Users will be asked whether to proceed when they try to delete a type node because deleting a type node will affect the nodes in the Type Graph whose types are the one to be deleted.

8.2 Type Graph

The editing operations applicable to a node in the type graphs include:

- Change name
- Change type
- Delete type
- Change flag
- Assign type to a child node

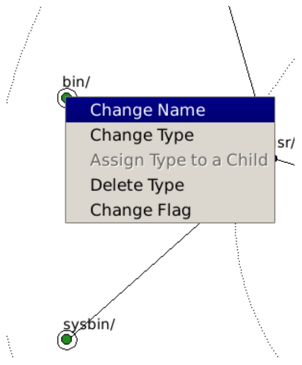


Figure 14: Context Menu of Type Node

All the editing operations can be accessed from a node's Context menu which pops up when right clicking on a type node, as shown in Figure 14.

The following rules apply to editing operations:

- The name of the root node cannot be changed as the root is always '/'.
- When the type of a node is deleted or the node has no type assigned, it will inherit the type of its parent until a new type is assigned through the "Change Type" option. The inheritance of a type is specified by the assign statement in the specification file with –r, for example in Figure 3, "assign –r generic_t /". The type of /etc will inherit the type of / in case that /etc has no type defined or the type of /etc is being deleted.
- When the type of a node is changed, the types of those children who inherit this node's type are also changed.
- Users can create a new type in the process of changing type, and a node representing the new type will be automatically added to the General Graph.

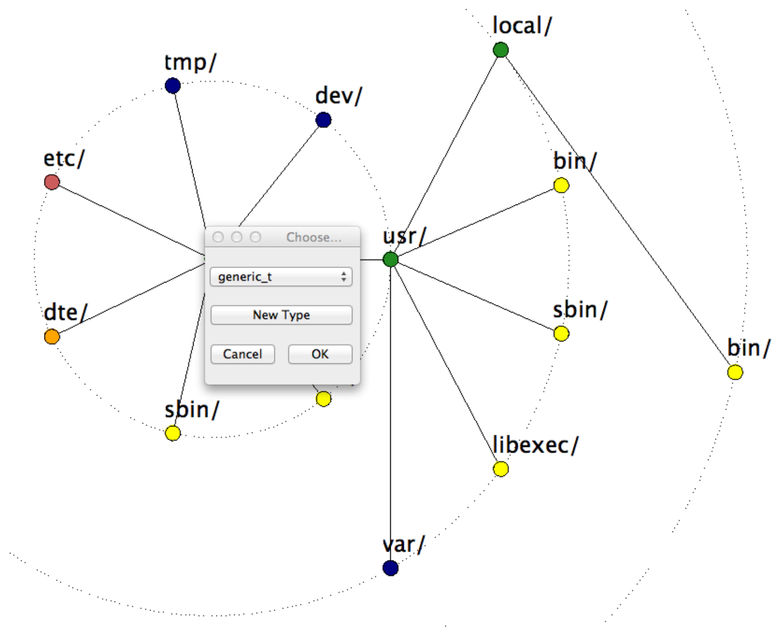


Figure 15: Change Type

Figure 15 depicts the context menu that appears after selecting "Change Type" from Context menu. Here the type of /usr is changed. Note that /usr/local/bin will take on this same type because it inherits the type of its parent.

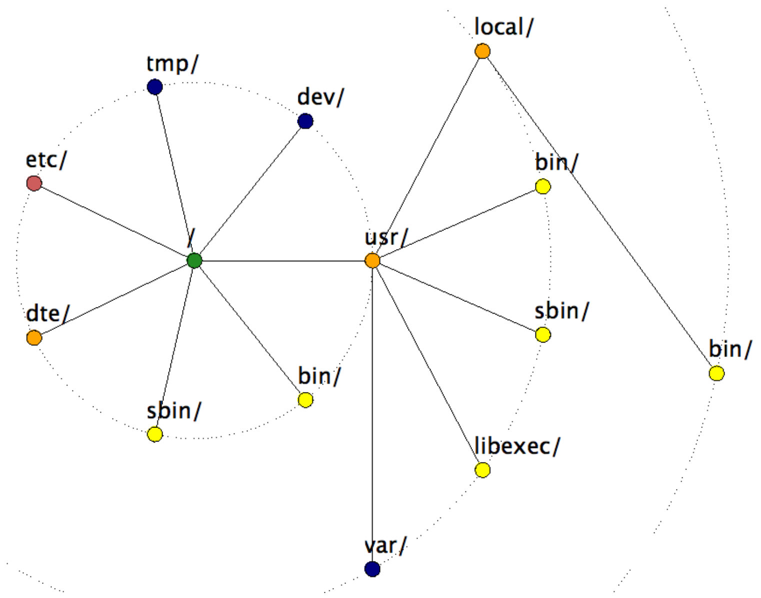
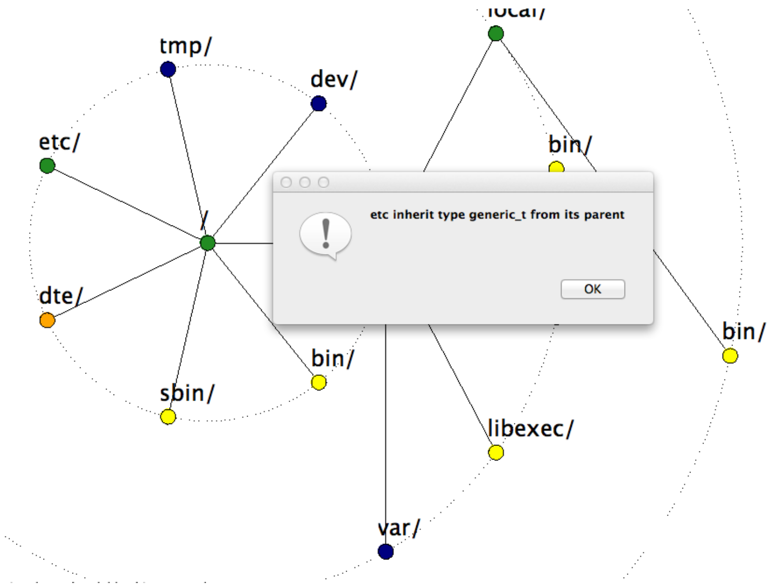


Figure 16: Delete Type



etc inherit type generic_t from its parent

Figure 17: Inherit Type

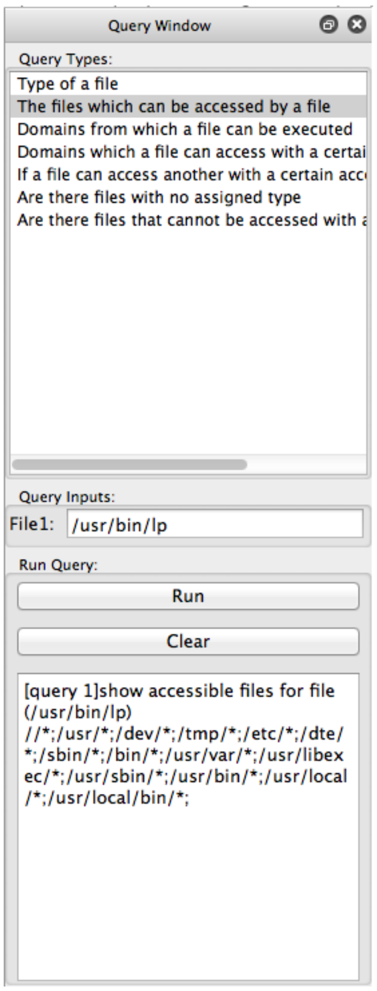
Figure 16 depicts deletion of a type. The type of /etc is changed to generic_t (the type of its parent) after its original type readable_t is removed (Figure 17).

If the type inherited is different from its original type, it is treated as if the node's type has been changed, and the types of its children may change.

Please note that the flag "–r" cannot be removed for those nodes with children (i.e., directories) because directories differ from files by the existence of the flag "–r". Users can add new nodes to a Type Graph by specifying its name, type and flag.

Again, new types can be created on-the-fly. There is no "delete a node" operation available because a node in this graph is a Directory or a File, which should not be removed in the Type Graph. Removing all types assigned to the node is equivalent to assigning "No Type" to that Node.

9 Analyze DTE Diagrams


Query Window
Query Types:
Type of a file
The files which can be accessed by a file
Domains from which a file can be executed
Domains which a file can access with a certai
If a file can access another with a certain acc
Are there files with no assigned type
Are there files that cannot be accessed with a
Query Inputs:
File1:
Run Query:
Run
Clear

File1: /usr/bin/lp

```
[query 1]show accessible files for file
(/usr/bin/lp)
//*;/usr/*;/dev/*;/tmp/*;/etc/*;/dte/
*;/sbin/*;/bin/*;/usr/var/*;/usr/libex
ec/*;/usr/sbin/*;/usr/bin/*;/usr/local
/*;/usr/local/bin/*;
```

Figure 18: Query Window

DTEvisual is able to carry out the following queries:

1) What is the type of a file?

2) What files can be accessed by a binary?

3) What files can be accessed by a binary in certain mode?

4) From which domains a file can be executed?

5) What domains can a file access with a specified permission?

6) Is a process created from a specified binary able to access another file with a specified permission?

7) Are there files without any type assigned?

8) Are there files which cannot be accessed with a certain permission?

9) What is the type of a file created from a domain?

10) Can a binary be executed from a specified domain?

In order to run a query, users must first bring up the Query Window (depicted in Figure 18) either by right-clicking in the empty visualization area or referring to the View Menu. After all the required inputs for a query have been given, the query is executed by hitting the Run button and the results appear in the Query Output.

Users can enable/disable or change the speed of the real-time animation that illustrates the answer to a query in the Settings Menu. The pause/resume button is only enabled after the animation begins. It is also possible to re-run a query quickly by double-clicking the query line (a line which contains "[query x]") in the output box. The animation of the query "Which files can be accessed by /usr/bin/lp" is shown below.

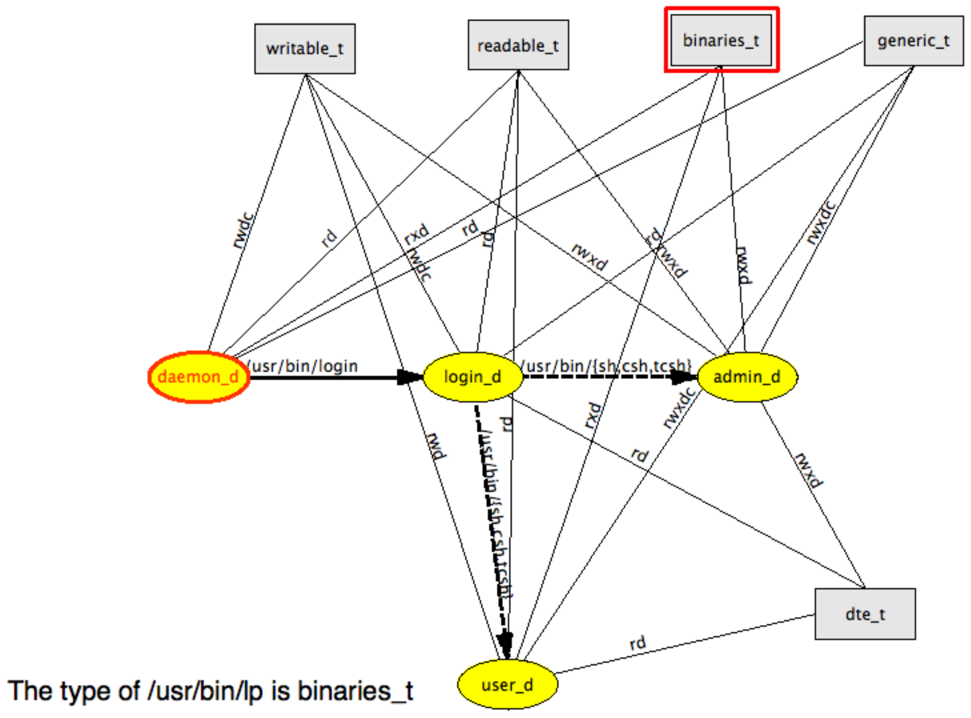
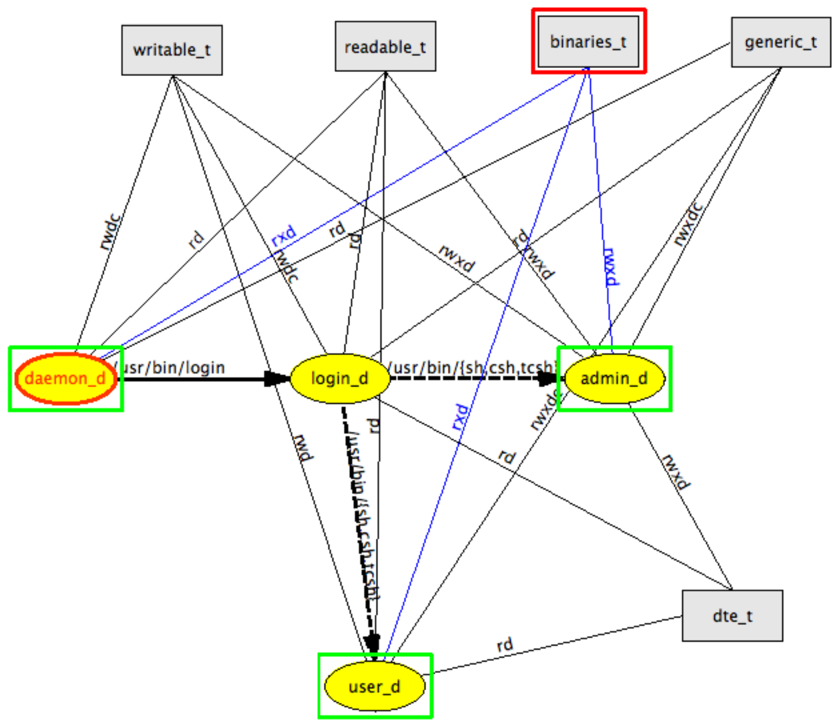
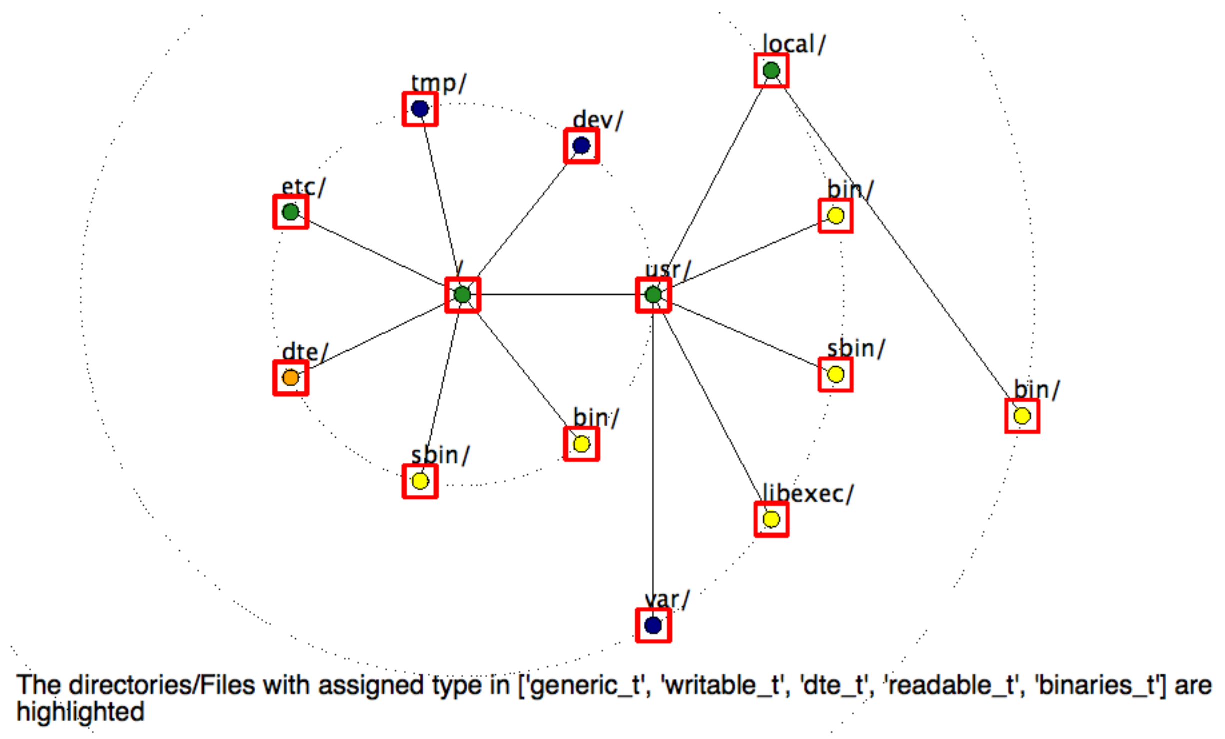


Figure 19: Determine the type of '/usr/bin/lp ', which is 'binaries_t'



Domain(s) ['admin_d', 'daemon_d', 'user_d'] have executable permission on type binaries_t

Figure 20: Find the domains that have executable permissions (x) on 'binaries_t'



The directories/Files with assigned type in ['generic_t', 'writable_t', 'dte_t', 'readable_t', 'binaries_t'] are highlighted

Figure 21: Object List

The Figure 20 shows all types that can be accessed from the domain found in the first step. Figure 21 lists all the objects whose types are among the types found in the second step.

Appendices

DTE specification normally contains 4 sections.

Appendix A

.1 Type Definition

Declares one or more object type names, which are then available to other parts of a DTE specification. The statement starts with keyword Type and then lists all the defined object types with suffix_t. No ordering is required.

Syntax format: "Type objecttype1_t, objecttype2_t, objecttype3_t, . . ."

For example:

Type dte_t, readable_t, generic_t, writable_t, sysbin_t, projectA_t

.2 Domain Definition

Domain definition is expressed as a list of tuples. Domain definition statement start with keyword domain and follows with domain ID and execution environment.

Defines a restricted execution environment composed of three parts:

1) Entry point programs, identified by pathname that a process must execute in order to enter the domain, for example, /usr/bin/login for domain login_d,
2) Access rights to types of objects, e.g. drwx->writable_t in domain daemon_d for domain_login_d. This contains multiple lines.
3) Access rights to subjects in other domains e.g. exec->user_d for domain login_d. Or allow Auto transition to another domain, like (auto->login_d) for daemon_d.

```
domain daemon_d = (  /sbin/init),
        (dr->generic_t,readable_t,dte_t),
        (drx->sysbin_t),
        (cdrw->writable_t),
        (auto->login_d);

domain login_d = (/usr/bin/login),
        (cdrw->writable_t),
        (dr->readable_t,generic_t,dte_t),
        (exec->user_d,admin_d,roleA_d,roleB_d,roleAB_d);

domain user_d = (/usr/bin/{sh,csh,ksh}),
        (drx->sysbin_t),
        (cdrwx->generic_t),
        (dr->readable_t,dte_t),
        (drw->writable_t);
```

.3 Initial Domain

This section contains one statement to declare the initial DTE domain.

initial_domain = daemon_d;

.4 Type Assignments

The assignments in this section will associate a type with one or more files.

Syntax format: "assign [-r] typeId_t directory1 directory2 . . . ;"

A statement may be recursive, in which case it applies to all files in the directory tree rooted at the named directory. Recursive assignment of a file with prefix P overrides an assignment for a file with a prefix shorter than P. For instance a recursive assign statement for /etc overrides a recursive assignment for /.

Below are more examples:

```
assign -r generic_t     /;
assign -r projectA_t    /projectA;
assign -r projectB_t    /projectB;
assign -r readable_t    /etc;
assign -r -s dte_t      /dte;
assign -r writable_t    /usr/var,/dev,/tmp;
```