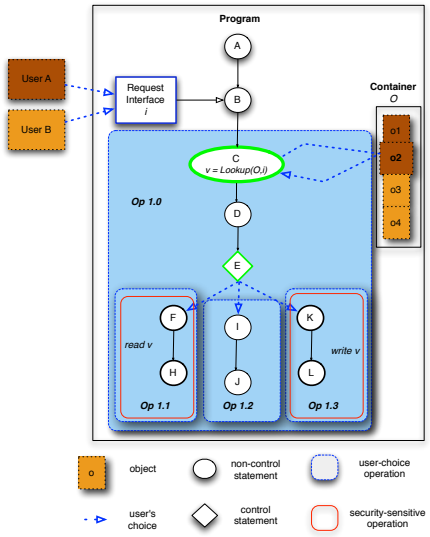
Leveraging 'Choice' to Automate Authorization Hook Placement

Divya Muthukumaran,Trent Jaeger, Vinod Ganapathy

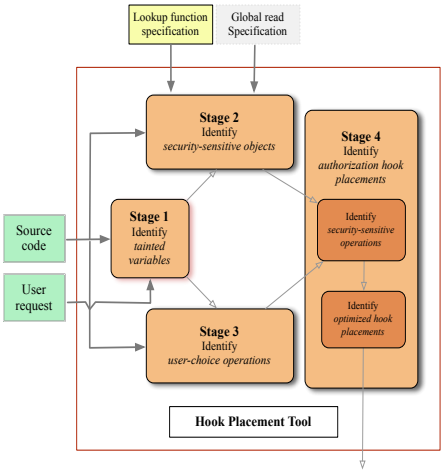
The Problem

- Servers **manage resources** on behalf of multiple, mutually-distrusting clients.
- Must interface with an **authorization policy** that determines whether a client request to access a resource is allowed.
  - Place **authorization hooks** to mediate all security-sensitive operations on shared resources.
- Largely **manual efforts** till date (X server,postgresql,gconf,dbus,etc)
  - Informal analysis of server code and discussions on developer forums.
  - Lack of consensus on basic concepts, such as the definition of **what constitutes a security-sensitive operation**.
  - No tool support to identify their occurrence in large code-bases.
- Prior efforts: Require training wheels that still necessitates **detailed understanding of the server's code base**.
- How do we automate this?

User 'Choice'



Design



Definitions

Tainted Variable: The set of tainted variables V_T (P) is the transitive closure of the relation data-dependence relation ($\mathscr{D}$) from the user request variables V_I(P).

Security sensitive object: A variable is security sensitive ($v \in V_S(P)$) if any following are true:

a) If it is assigned a value from a container via a lookup function using a variable $v \in V_T(P)$,

b) If $\mathscr{D}$ is true for some $v' \in V_S(P)$.

c) If it is a global variable and in the set $V_T(P)$.

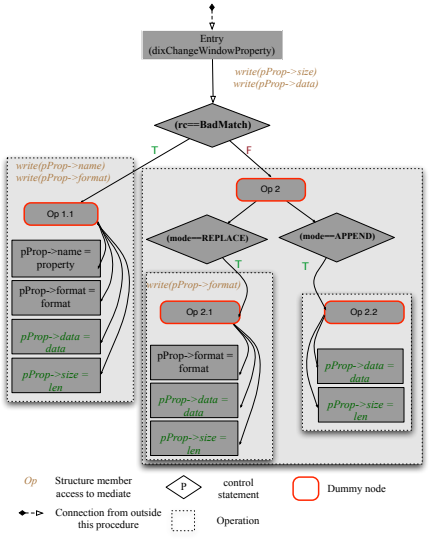
Operation: An *operation* is a subgraph of the Control Dependence Graph rooted at dummy node. If a dummy node's control statement is predicated on a variable in $V_T(P)$, then the operation rooted at the dummy node is a user-choice operation

Security sensitive operation: A operation is security-sensitive if it is a user-choice operation and it allows the user to access a variable in V_S (P).

Example

```
int ChangeWindowProperty(ClientPtr *c,
    WindowPtr * w, int mode)
{
  WindowPtr * win;
  PropertyPtr * pProp;
  err = LookupWin(&win,stuff->window, c);
  rc =  LookupProperty(&pProp, win, stuff >property, c);
  if (rc == BadMatch)
    {/* Op 1*/
     pProp->name = property;
     pProp->format = format;
     pProp->data = data;
     pProp->size = len;
    }
  else
    {/* Op 2 */
     if (mode == REPLACE)
      { /* Op 2.1 */
        pProp->data = data;
        pProp->size = len;
        pProp->format = format;
      }
     else if (mode == APPEND)
      {/* Op 2.2 */
        pProp->data = data;
        pProp->size += len;
      }
    }
}
```

Hook Placement



Results

Program	X Server	postgres	pennmush	memcached
LOC	28k	49k	78k	9k
Total variables	*7795*	*12350*	*24372*	*2350*
Tainted variables	2975 (38%)	5100 (41%)	4168 (17%)	490 (20%)
Security sensitive variables	823 (10%)	402 (3%)	1573 (6%)	82 (3%)
Data Structures	*404*	*278*	*311*	*41*
Sensitive Data structures	61 (15%)	30 (10%)	38 (12%)	7 (17%)
User-choice Operations	*4760*	*5063*	*6485*	*996*
Sensitive operations	1382 (29%)	1378 (27%)	1382 (21%)	203 (20%)
Hooks	532 (11%)	579 (11%)	714 (11%)	56 (5%)

Supported by NSF CNS-0905343 and HP Labs IRP