Foundations of Computer Security
Lecture 13: Covert Channels I

Dr. Bill Young
Department of Computer Sciences
University of Texas at Austin
Consider the simple lattice of labels in the diagram, where $H > L$. There are no need-to-know categories in this system.

If this represents a BLP lattice, then information flow is permitted from L to H, but not vice versa. This captures the metapolicy of this simple system.

If we can instantiate this system such that BLP is satisfied, but information flows in violation of the metapolicy, something is clearly wrong.
Consider a simple system that has READ and WRITE operations with the following semantics:

**READ (S, O):** if object O exists and $L_S \geq L_O$, then return its current value; otherwise, return a zero.

**WRITE (S, O, V):** if object exists O and $L_S \leq L_O$, change its value to V; otherwise, do nothing.

These operations pretty clearly are acceptable instances of READ and WRITE for a BLP system.
Suppose we want to add two new operations, CREATE and DESTROY to the system, with the following semantics:

**CREATE (S, O):** if no object with name O exists anywhere on the system, create a new object O at level $L_S$; otherwise, do nothing.

**DESTROY (S, O):** if an object with name O exists and the $L_S \leq L_O$, destroy it; otherwise, do nothing.

These operations seem to satisfy the BLP rules, but are they “secure” from the standard of the metapolicy? Why or why not?
In this system, a high level subject $S_H$ can signal one bit of information to a low level subject $S_L$ as follows:

<table>
<thead>
<tr>
<th>$S_H$ Transmits 0</th>
<th>$S_H$ Transmits 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create ($S_H$, F0)</td>
<td>do nothing</td>
</tr>
<tr>
<td>Create ($S_L$, F0)</td>
<td>Create ($S_L$, F0)</td>
</tr>
<tr>
<td>Write ($S_L$, F0, 1)</td>
<td>Write ($S_L$, F0, 1)</td>
</tr>
<tr>
<td>Read ($S_L$, F0)</td>
<td>Read ($S_L$, F0)</td>
</tr>
<tr>
<td>Destroy ($S_L$, F0)</td>
<td>Destroy ($S_L$, F0)</td>
</tr>
</tbody>
</table>

In the first case, $S_L$ sees a value of 0; in the second case, $S_L$ sees a value of 1. Thus, $S_H$ can signal one bit of information to $S_L$ by varying its behavior.
Who cares if one bit flows from high to low?

- It’s enough to show that BLP cannot guarantee that the metapolicy is satisfied.
- If $S_L$ and $S_H$ can coordinate their activities, $S_H$ can transfer arbitrary amounts of information to $S_L$, given enough time.

In an access control policy like BLP, objects are the only entities recognized to carry information.

For the channel above, the “information” is not in the contents of any object. It’s in the answer to the question: can $S_L$ read an object named $O$?
If $S_L$ ever sees varying results depending on varying actions by $S_H$, that could be used to send a bit of information from $S_H$ to $S_L$, in violation of the metapolicy.

Such a mechanism is called a *covert channel*. 
Lessons

An access control policy constrains information flowing by subjects reading or writing objects.

There may be other system features that could be manipulated to convey information.

Such channels are called “covert channels.”

Next lecture: Covert Channels II