Graph cuts can be used to solve security problems
Divya Muthukumaran, Dave King and Trent Jaeger

The Problem

We need a common framework to reason about security problems in programs and policies. Consider three security problems:

a) **Mediation Placement in programs:** Many programs make runtime security decisions about whether to accept untrusted inputs, provide access to program data, or perform operations on the behalf of request. Programmers therefore need to insert mediation statements viz., declassifiers, endorsers, authorization hooks to ensure that at runtime, the behavior of the program is in accordance with the security policy expressed by the type system and labels. How do we place such mediation statements automatically?

b) **Privilege Separation:** Privilege separation divides a program into two parts, a privileged monitor that performs the privileged operations and an unprivileged slave that faces the potential adversary. The challenge in privilege separation is to identify where to split the code.

c) **Policy Error Resolution:** Designing access control policies involves choosing accesses that allow function without compromising security requirements. Oftentimes policies get very complex and end up authorizing unintended flows. Such flows indicate policy errors that need to be resolved.

Why graph-cuts?

- Security problems arising from information-flow errors can be modeled as a graph cut problem
- Modeling security problems as graph-cut problems has the following steps.

![Graph cuts and security problems](https://example.com/graph-cuts-security-problems.png)

- Example: The figures show the code snippet and the corresponding information flow graph with the graph-cut based mediation placement suggestions for `logrotate`

Challenges

- **Identifying sources and sinks:** Sources and sinks of sensitive information have to be identified. How do we automate this?
- **Cut-conjunction problems:** Extending the graph cut solution to work on general security lattices with multiple sources and sinks. Preliminary solution uses a greedy algorithm that solves the generalized Hitting Set problem.
- **Converting cuts to security solutions:** The programmer needs to translate graph cuts to the appropriate security solutions. For example, placing mediation statements at the cut locations. For privilege separation, splitting the code at the cut location and creating the master and slave. Depending on the security problem the cuts may need to be customized.

Tool for C programs

- We currently have a graph cut tool to solve the mediation placement problem in JAVA programs. The mediation locations suggested by this tool matches manual mediation placement 80% of the time.
- But we found that there are many legacy C programs that need this kind of mediation. SELinux/MLS identified 54 programs that are trusted by SELinux to enforce MLS on their execution. We are working on extending the tool to work on C programs.
- We intend to extend these tools to solve the other two problems as well.