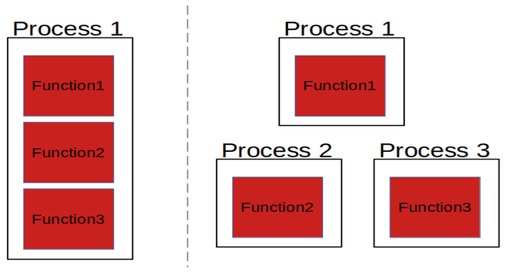
A Case Study of Fine-Grained Software Compartmentalization using cURL

Stephen Carrasquillo | Junyong Zhao | Henry Zhu | Nik Sultana | Boon Tau Loo
University of Pennsylvania

Background



Compartmentalization breaks up functions into separate processes

LibCompart is a library currently being developed to make compartmentalization easier.

Compartmentalization is useful for information security since it li**mits access** to sensitive information.

cURL is an **open source** file transfer utility.

It is widely distributed in many Linux distributions.

Motivation

- ✔ Create a functioning version of cURL with compartmentalization using LibCompart.
- ✔ Examine the impact of compartmentalization on cURL's execution.

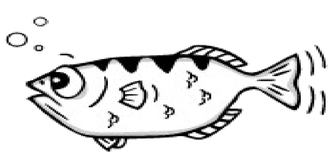
Methodology

Identify the critical functions

Use **GDB** to step through cURL.

-

Key Functions are identified where the curl **interface** utilizes **instances** of curl-lib.

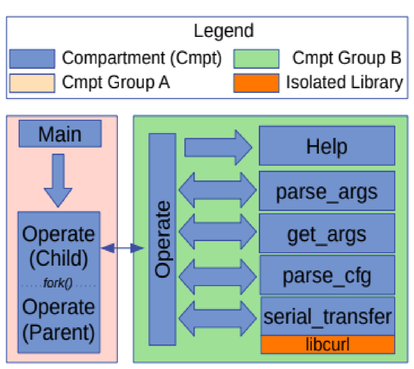


Isolate functions

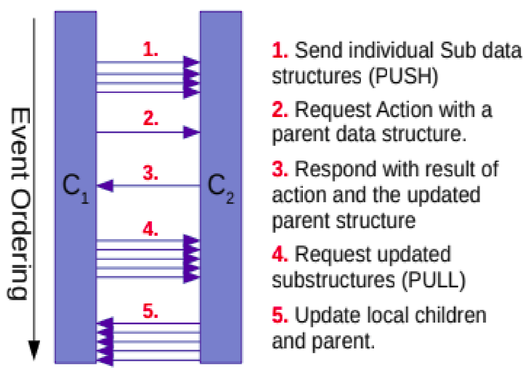
Use **LibCompart** to assign functions to compartment calls.

-

Replace functions in source code with **compartment calls.**



Serialize Data Structures



Use the **serializer** tool to create stubs for marshalling and unmarshalling primitive data structures.

-

Create **synchronization** protocol for complex relationships.

Test and Validate Results

Create 2 test cases:

- Download of a simple web page (www.google.com) using cURL's HTTP module.
- Download of a 65MB video file from a local network server using cURL's FTP module.

- Both Tests were performed 10X using a bash script to record the start and end time.

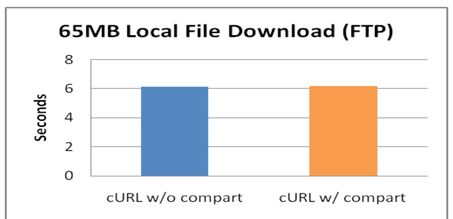
Results / Findings

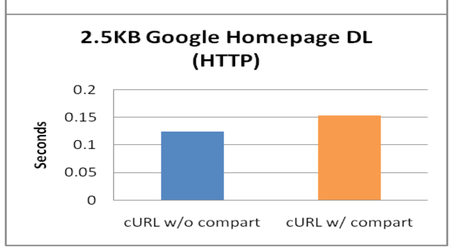
Performance:

- Compartments added on average 6ms to large data transfers and 30ms to smaller file downloads
- More testing and validation is required to understand the full impact of compartmentalization.

File Accuracy:

- Some changes were reported on the google web page download; However, these changes were expected as they are unique hashes for each page visit.
- The 65MB file reported no difference in cURL's the compartmentalized version.





Final Thoughts

- Compartmentalization was successfully achieved with LibCompart in cURL's source code.
- While successful there are still challenges that need to be overcome.
- More compartmentalization efforts will be required to fully tackle these challenges.

Contact: carste@seas.upenn.edu
Further information: https://cobre.cis.upenn.edu/