A Model for Securing Application Chains in a Cloud

Donghoon Kim and Mladen A. Vouk
{dkim2, vouk}@ncsu.edu

Goals & Assumptions

Explore a model for securing software application chains in a cloud. Applications chains are controlled by a workflow engine.

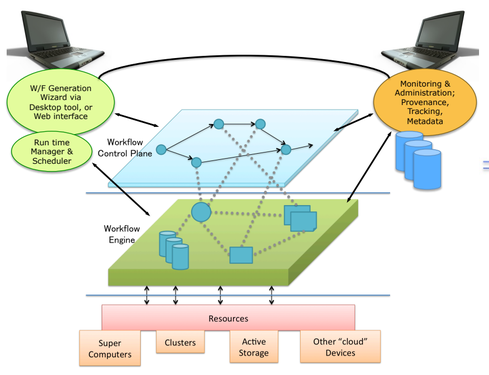
- Assumption1: Individual software application chain elements are reliable and secure under normal operational profile.
- Assumption2: Attacks are based primarily on non-operational profiles
- Assumption3: Assuring normal operational profile (OP) by securing access and individual input/output data (flow) integrity ensures security.

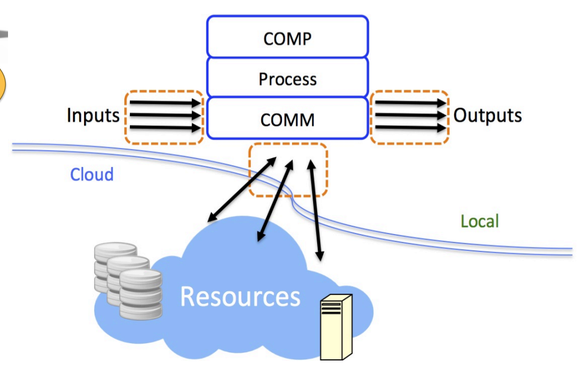
Approach

- A software application chain is represented by a directed graph on a data-flow control plane
- A simple serial (sequential) workflow example

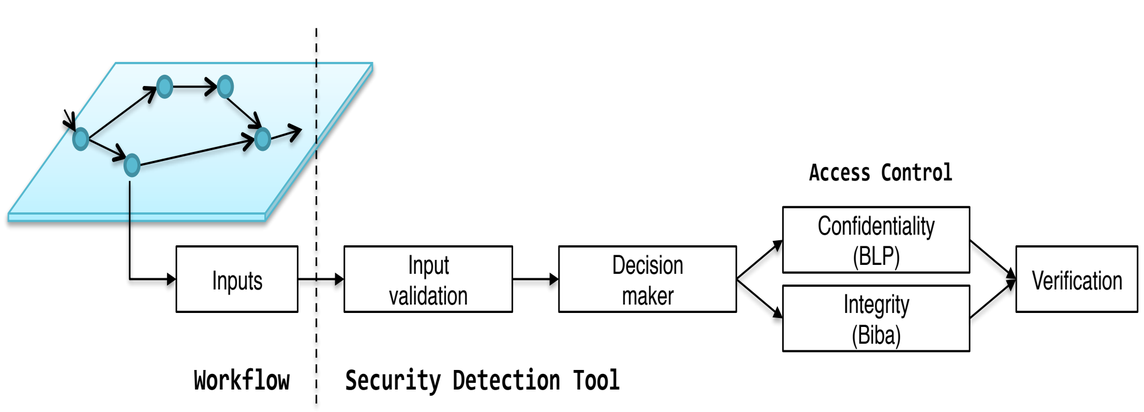
$$x_1 \xrightarrow{f_1} y_1(=x_2) \xrightarrow{f_2} y_2 \ldots (=x_{n-1}) \xrightarrow{f_{n-1}} y_{n-1}(=x_n) \xrightarrow{f_n} y_n$$

- x_1: 1st input, y_n: the final output
- Every x_i (i.e., $i = 1, \ldots, n$), $x_i \in OP_i$
- OP_i is the "security safe" operational profile of f_i i.e., a set of expected inputs into component f_i
- x_n and y_n should be verified





- Each application (workflow/chain component) may have elements:
  - COMP (Computation), Process, COMM (Communication), Input (Dataflow input), Output (Dataflow output)



- Fundamental validation process for a dataflow model
  - Provenance data and security analysis of provenance data validates process and OP
  - For example: input validation may be done using whitelist, blacklist, regular expressions, etc.

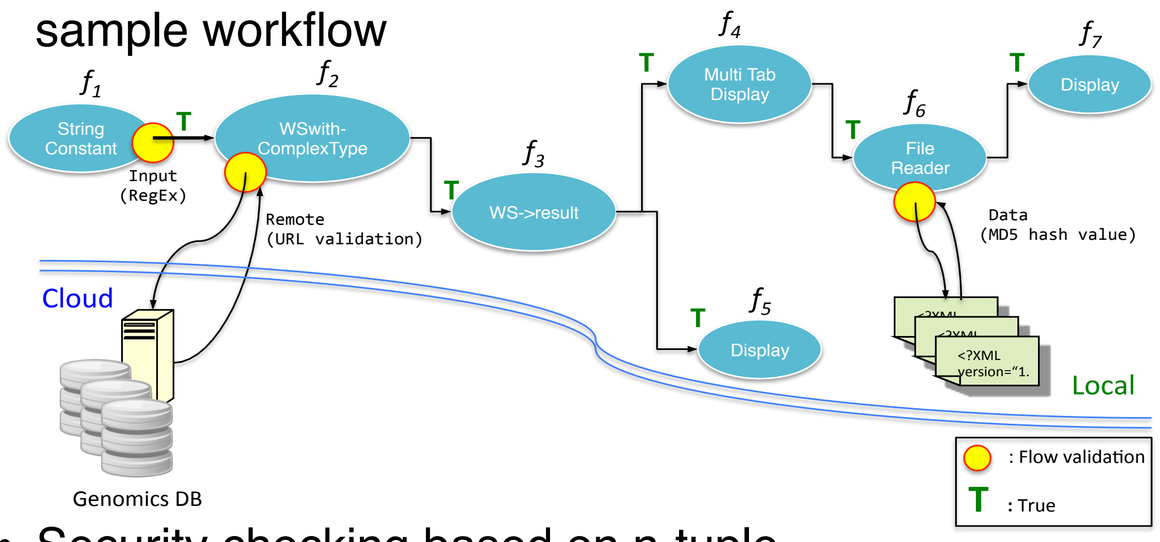
Let n-tuple ***WM =<F, O, C, SP, SC >*** describe the workflow and its security properties

- *F* : a set of operations (functions, processes, transformations), e.g.,$\{f_1, f_2, \cdots, f_n\}$
- *O* : objects; data, data objects and flows
- *C* : connectivity matrix for directed graphs describing the workflow,
  loops are allowed, loop stopping criteria are in the transformation node f .
- *SP* : security property, e.g., {Input, Remote, Data}
  (1) Input validation
  (2) Remote access validation, security attestation, remote input/out
  (3) Data integrity
- SC : security class, e.g., {Secure, Insecure}, but could be multilevel - e.g., Top Secret, Secret, Insecure.

Case Study

Workflow validation

- A security assessment of dataflows with **Kepler** sample workflow



- Security checking based on n-tuple

	Input	Remote	Data	3 bit	⊕	Secure
f_1	1	0	0	100	0	100
f_2	0	1	0	010	0	010
f_3	0	0	0	000	0	000
f_4	0	0	0	000	0	000
f_5	0	0	0	000	0	000
f_6	1	0	1	101	0	101
f_7	0	0	0	000	0	000

Discussion

The proposed model provides an overall framework for securing software application chains

- This approach assumes that (1) attacks are limited to abnormal data chains on Input, remote access, and output channels (or flows).
- Idea is to limit I/O interactions to normal operational profile and validate the flows using operational profile or certification based signals
- Model may be a good way of protecting from zero-day attacks. For example, deserialization vulnerability:

  "WebLogic Server component allows remote attackers to execute arbitrary commands via a crafted serialized Java object."