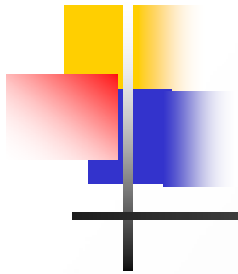
Attack Model: Graph based Attack Representation and Taxonomy

Vasily Desnitsky and Igor Kotenko

Computer Security Research Group,
St. Petersburg Institute for Informatics and
Automation of Russian Academy of Sciences

Contents

1. Introduction
2. Graph based attack representation
3. An example of program tampering attack
4. Advantages of suggested attack representation
5. Estimation of attack realization time
6. Preliminary attack model taxonomy
7. Conclusion

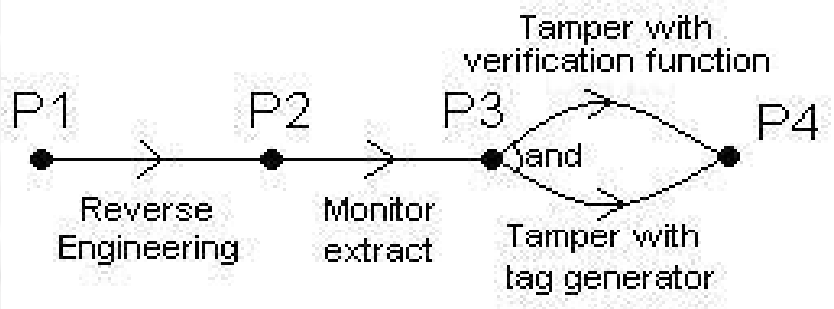


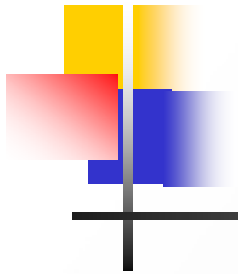
Introduction

- It is important to find a formal presentation of attack
- Our approach is based on oriented graph representation which uses program state notion and action one
- There are a lot of graph and tree based attack model representations in computer security field. Our model is aimed to take into account ReTrust specific features

Graph based attack representation (1/3)

- Attack is presented by oriented graph
  - Vertexes – states of target program
  - Arcs – adversary actions on the target program
- Attack model has
  - Initial state – an initial untampered program
  - Final state – a broken one
- Graph may have two types of branches
  - OR – it is sufficient to execute one option only
  - AND – it is needed to execute all options





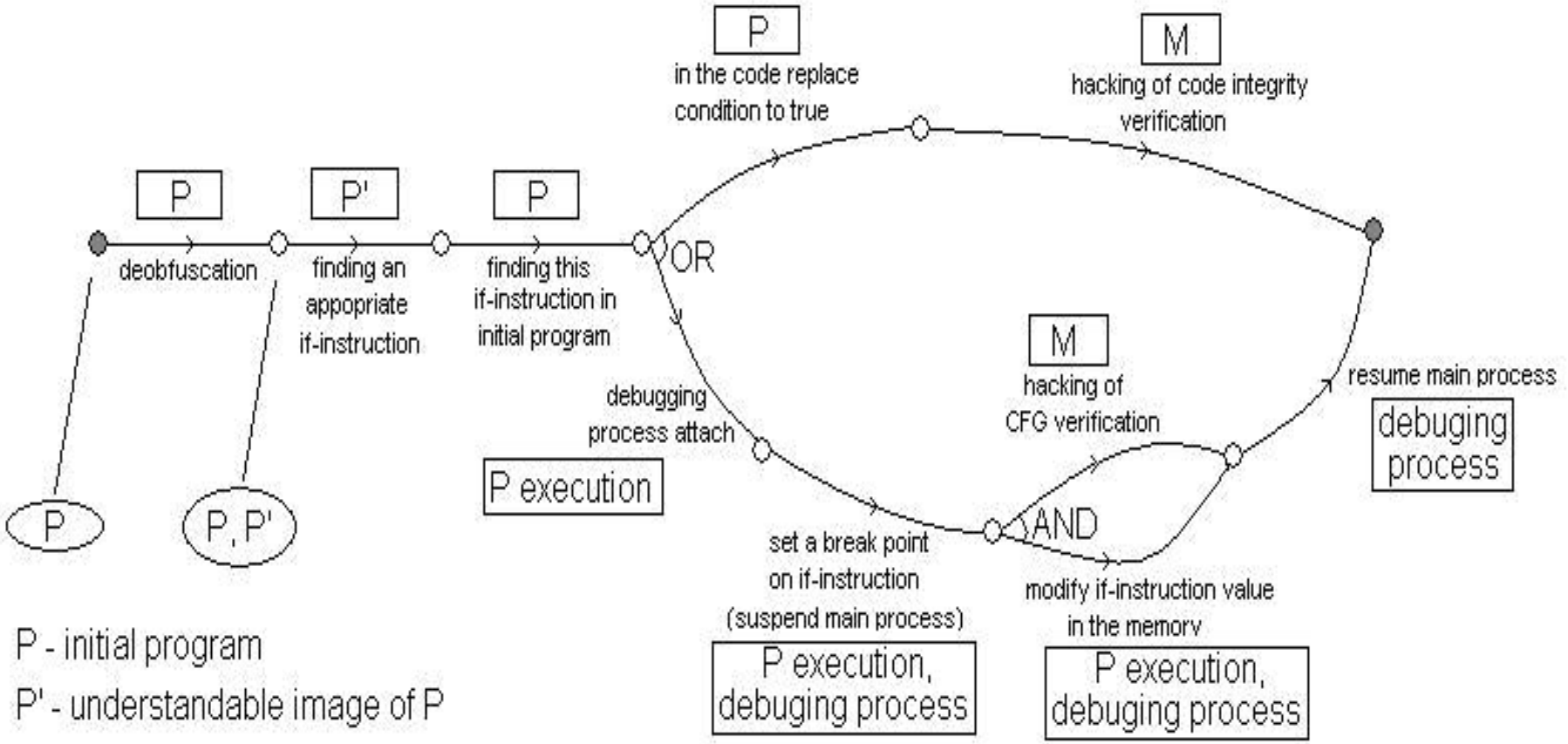
Graph based attack representation (2/3)

- Program state is described by
  - Program attributes, e.g.
    - (non-), (de-) obfuscated
    - Presence or absence of a secret key in a program
  - Additional objects and data extracted from the program earlier or from outside, e.g.
    - Monitor, signature generator, secret key
  - Modified program parts or modules and modification type

Graph based attack representation (3/3)

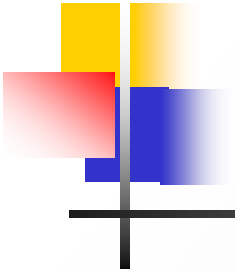
- Actions
  - Action description
  - A set of objects which a the subject of action
- Two kinds of actions
  - Modification of program or its part
    - e.g. deobfuscation, code modification, embed debugging process, etc
  - Analysis of program or its part
    - e.g. search of specific code instruction, monitor analysis
- An action may be detailed to some set of sub-actions
  - e.g. concrete reverse engineering techniques, extracting monitor methods, specific modification methods, etc

An example of program tampering attack



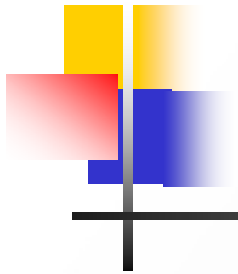
Advantages of suggested attack representation

- Attack representation obviousness
- Demonstration of action relationships In time
  - e.g. possible parallelism of actions
- It helps to estimate computational complexity of an attack fulfillment
- It helps to reveal the way of attack effectiveness, e.g.
  - Parallel execution of several actions
  - To eliminate repeated fulfillment of already executed intermediate computation or same data search



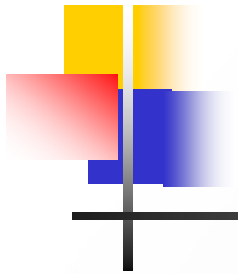
Estimation of attack realization time

- Estimation of machine time for breaking the certain program component or the execution of some actions
- Human factor



Preliminary attack model taxonomy

- Attack
  - Attack graph
    - program states
      - program attributes
      - additional objects and data
    - actions
      - modifications
      - analysis
    - AND/OR branches
- General attack classification
- Attackenv
- Attacktext
- Attackload
- Attackrun
- Main kinds of attacks
  - Riverse engineering attack
  - Clonning attack
  - Differential analysis attack
  - Separation attack



Conclusion

- In the future:
  - To extend and detail the introduced notions of attack models
  - To create the complete attack model taxonomy
- And in particular
  - To detail the notion of program state
  - To develop methods to estimate computational complexity of attacks
  - How to estimate human factor?