First Principles Vulnerability Assessment

James A. Kupsch
Barton P. Miller
Computer Sciences Department
University of Wisconsin

Elisa Heymann
Eduardo César
Computer Architecture and
Operating Systems Department
Universitat Autònoma de Barcelona

Cloud Computing Security Workshop 2010 (CCSW'10)
Chicago, IL, USA
October 8, 2010

Motivation

- We started by trying to do something simple:
  - Increase our confidence in the security of some critical grid middleware
- We ended up developing a new manual methodology:
  - *First Principles Vulnerability Assessment (FPVA)*
- We found some serious vulnerabilities … and more vulnerabilities … and more.

First Principles Vulnerability Assessment

- Manual assessment process – analyst centric
- Insider – have access to
  - Developers
  - Source code
  - Documentation
- Independent from development team
  - No agenda
  - No blinders
- First Principles – let the process guide the search

FPVA: 4 Step Process

1. Architectural Analysis
2. Resource Analysis
3. Trust and Privilege Analysis
4. Component Analysis

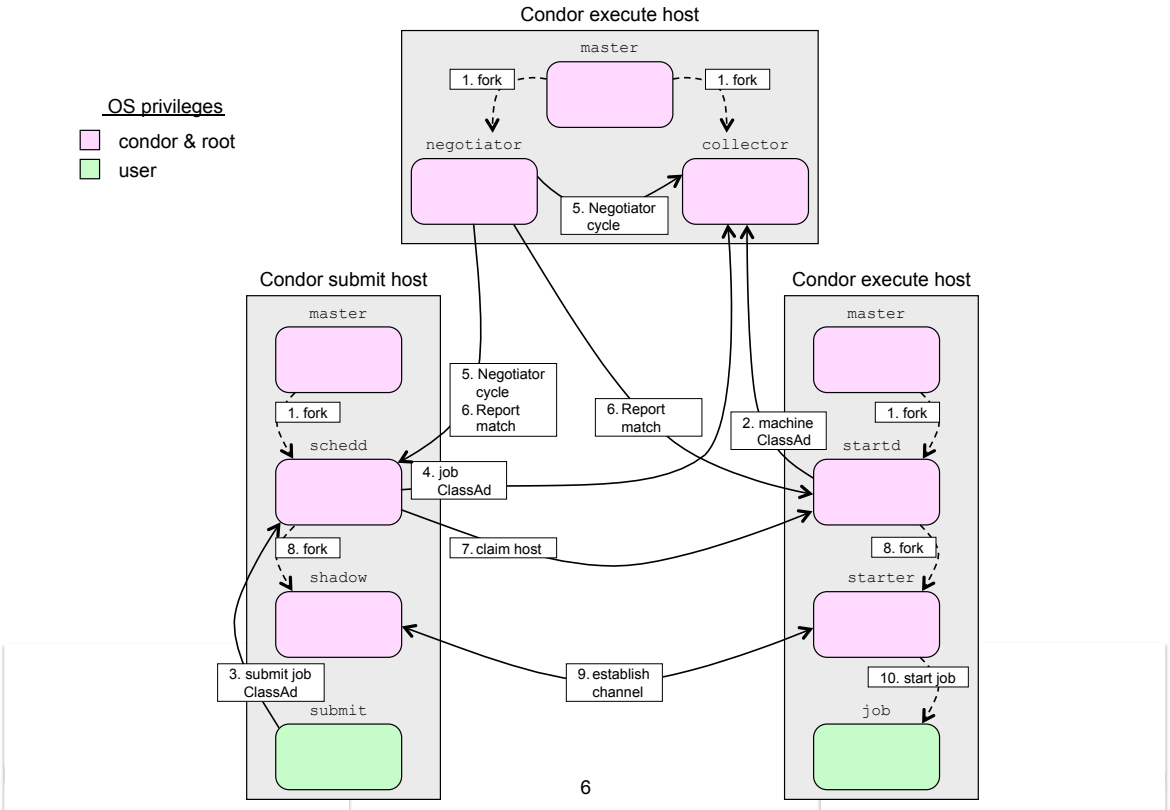
Post-FPVA Activities:

- Disseminate vulnerability reports to developers with suggested remediation
- Council developers about fix, disclosure and security release process

FPVA: Steps 1 - 3
Understanding the System

1. **Architectural Analysis** – functionality and structure of the system, major components, communication channels
2. **Resource Analysis** – Objects in the system and allowed operations
3. **Trust and Privilege Analysis** – trust boundaries of components, privilege model presented to users, and external privilege systems used

Condor Job Submission



FPVA: Step 4
Searching for Vulnerabilities

- Connect user supplied data to security violation of a resource
- Audit the source code
- Guide search using
  - Previous analyses and diagrams
  - Knowledge of how vulnerabilities arise
    - Dangerous functions
    - Dangerous idioms

Systems Assessed

Condor	**Condor**, University of Wisconsin Batch queuing workload management system 15 vulnerabilities 600 KLOC of C and C++
SRB	**SRB**, SDSC Storage Resource Broker - data grid 5 vulnerabilities 280 KLOC of C
MyProxy	**MyProxy**, NCSA Credential Management System 5 vulnerabilities 25 KLOC of C
NIKHEF	**gLExec**, Nikhef Identity mapping service 5 vulnerabilities 48 KLOC of C
Open Science Grid Fermilab	**Gratia Condor Probe**, FNAL and Open Science Grid Feeds Condor Usage into Gratia Accounting System 3 vulnerabilities 1.7 KLOC of Perl and Bash

Systems Assessed (cont.)

	**Condor Quill**, University of Wisconsin DBMS Storage of Condor Operational and Historical Data 6 vulnerabilities 7.9 KLOC of C and C++
	**Condor Privilege Separation**, University of Wisconsin Restricted Identity Switching Module 2 vulnerabilities 21 KLOC of C and C++
	**VOMS Admin, INFN** Web management interface to VOMS data (role mgmt) 4 vulnerabilities 35 KLOC of Java and PHP
	**CrossBroker**, Universitat Autònoma de Barcelona Resource Manager for Parallel and Interactive Applications 2 vulnerabilities 97 KLOC of C++

Improving FPVA

FPVA requires a costly asset:
a skilled security assessor

Can existing tools reduce the cost of manual assessment?

What can the tools find?

Case Study

Goal is to study the best tools out there.
Apply them to a system we studied.
Use our results as a ground truth.

- Talked to academics, military, and industry people about what they thought were the best tools:
  - Coverity Prevent
  - Fortify SCA
- Review tool output
  - Defect with matching location of known vulnerability is a positive result
  - Sample tool output to understand results

Ground Truth:

FPVA Condor Results

15 significant vulnerabilities discovered

`http://www.cs.wisc.edu/condor/security/vulnerabilities`

- 7 implementation bugs
  - easy to discover - localized in code
  - use of troublesome functions:
    `exec`, `popen`, `system`, `strcpy`, `tmpnam`
- 8 design flaws
  - hard to discover in code - higher order problems
  - defects include:
    - injections, directory traversals, file permissions, authorization & authentication, and a vulnerability in third party library

Results of Static Analysis Tool Study of FPVA Vulnerabilities in Condor

	Coverity	Fortify SCA	
Defect Reports:	2,986	15,466	total
		3	critical
		2,301	hot
		8,101	warm
		5,061	info
Defect Categories:	70	45	
FPVA Vulnerabilities Found:	1	6	total
	1	6	impl. bug
	0	0	design flaw

Tools: The Good and the Bad

Good:

- Easy to use
- Finds some simple implementation security problems
- Finds many minor security problems such as resource leaks
- Finds questionable programming practices

Bad:

- Reports false defects - False Positive problem (large number is overwhelming)
- Misses real vulnerabilities - False Negative problem
- Requires skilled operator to understand output

Research Directions

- Can we automate the discovery of some of the vulnerabilities not found by current tools?
- Can we automate some of the architecture, resource, and trust and privilege analyses using static or run-time analysis of the system?

Questions

For more information see:

http://www.cs.wisc.edu/mist