Security Risks in Clouds and Grids

Elisa Heymann

Computer Architecture and
Operating Systems Department
Universitat Autònoma de Barcelona

Elisa.Heymann@uab.es

Barton P. Miller
James A. Kupsch

Computer Sciences Department
University of Wisconsin

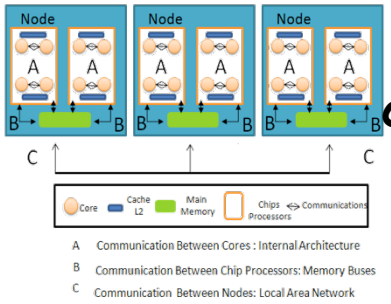
bart@cs.wisc.edu

Condor Week
May 5, 2011

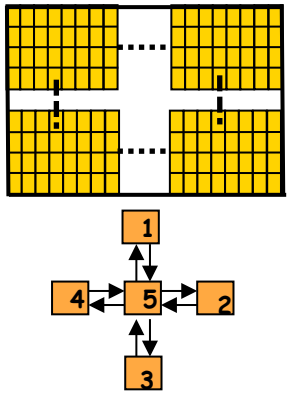
Efficient execution of SPMD Applications on Multicore Environments

Problem

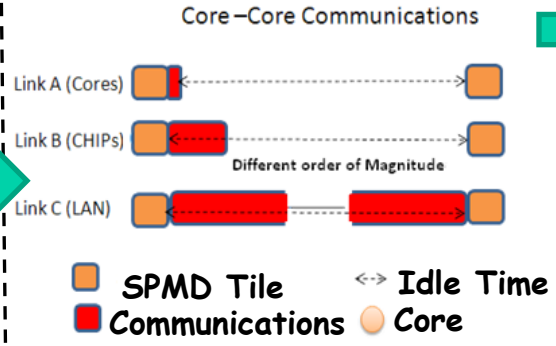
Multicore Environment



Hierarchical communication architecture



SPMD Application Message Passing Interface

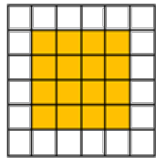


Objective

- Maximum Speedup
- Efficiency over a defined threshold

How?

Methodology



E Edge tile

I Internal tile

SuperTile allows to overlap the internal computation with edge communication

→ **Ideal Size of Supertile**

→ **Ideal Number of Core**

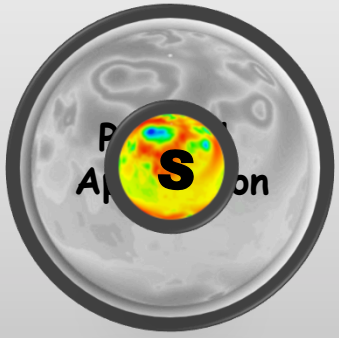
THE UNIVERSITY of WISCONSIN MADISON

Universitat Autònoma de Barcelona

NATO OTAN

NSF

PARALLEL APPLICATION SIGNATURE FOR PERFORMANCE PREDICTION



Instrumentation

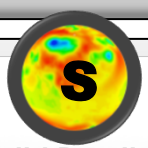
Base Machine

Collection data

Parallel Application Model

Patterns Identification

Extract Phases and Weights



Build Parallel Application Signature

(Coordinated Checkpoint + Phases + Weights)

Time of each Phase by weight

Prediction

Target Machine A

BT-256	128	98.52%	1.5%
SP-256	128	99.23%	6.4%
SMG2000-256	128	98.25%	3.8%
Sweep3D-256	128	92.38%	3.5%

Time of each Phase by weight

Prediction

Target Machine B

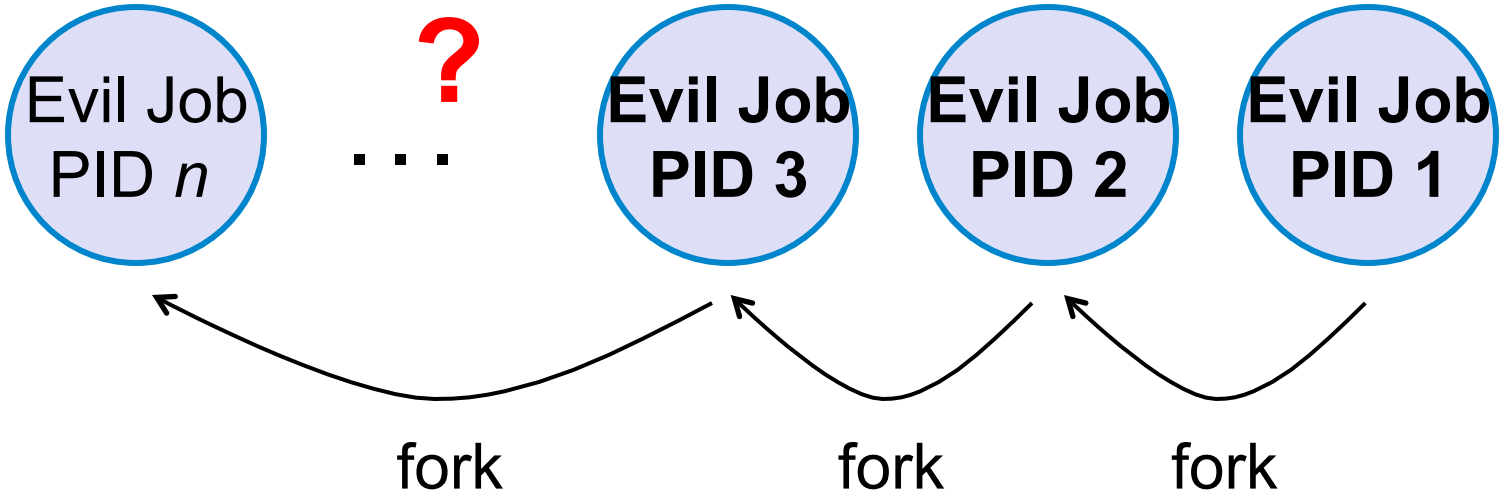
BT-256	256	98.63%	6.4%
SP-256	256	99.37%	3.4%
SMG2000-256	256	98.24%	3.8%
Sweep3D-256	256	92.35%	6.2%

Possible Threat?

- Clouds and Grids have databases with management and operational information
- Denial of Service:
  - Prevent updates in the database

Possible Threat?

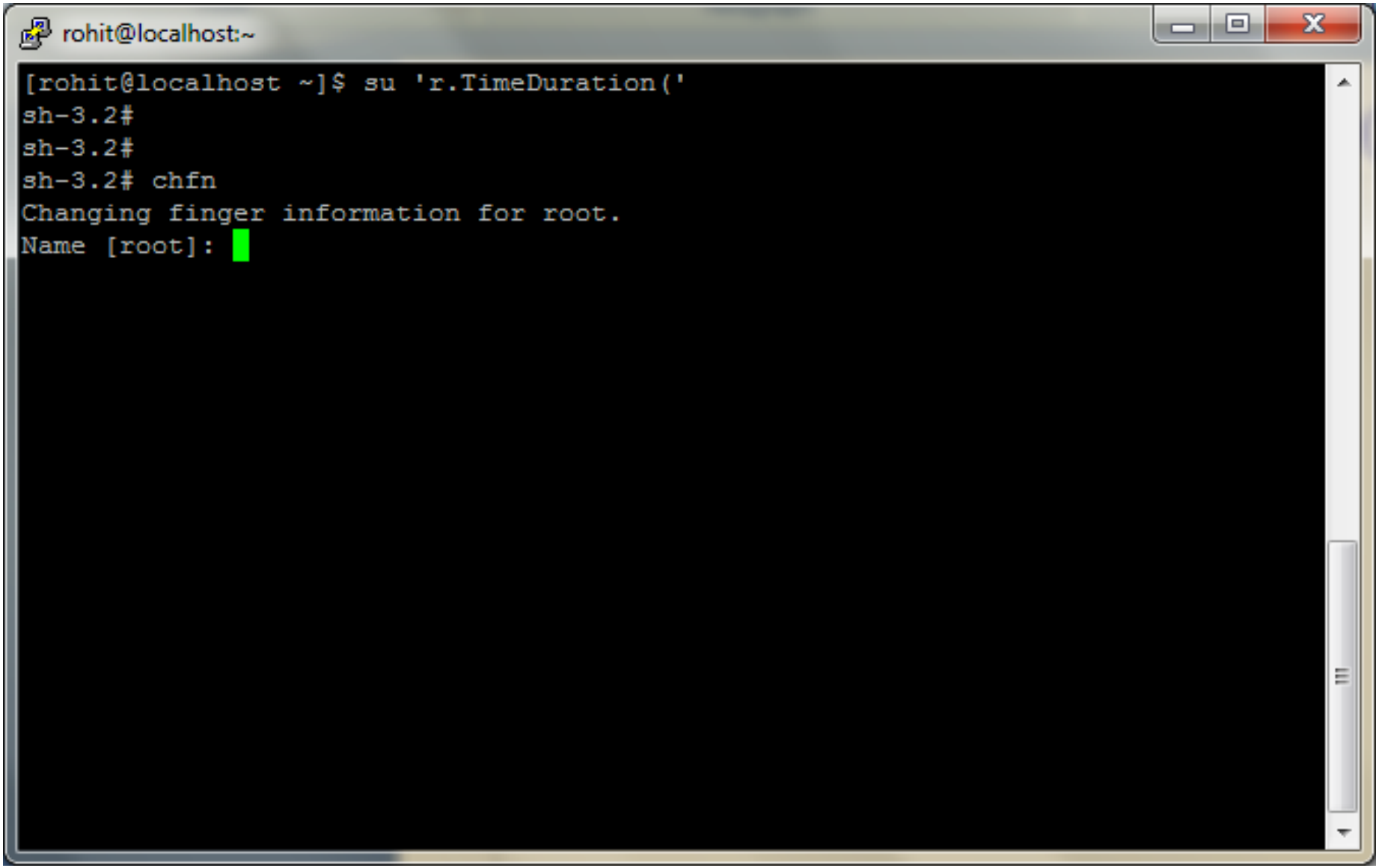
- Hijack machines
  - Process escapes Cloud/Grid/control: Keeps forking and exiting to escape detection.



Possible Threat?

- Cloud/Grid Accounting System
  - Maintains a Grid-wide view of resource utilization.
    - Job Submission (Priority in the batch queue, CPU time, Memory usage)
    - Storage (Disk usage, Tape storage)
  - Accounting Information ***easily*** available to people (web interface) and to applications (Web Services)
- Use the Accounting System for bad purposes.

Possible Threat?



Real Threat!

GRATIA-CONDOR-2010-0003

Summary:

Any user that can submit Condor jobs on the host running Gratia Condor probe, can execute arbitrary code as the root user.

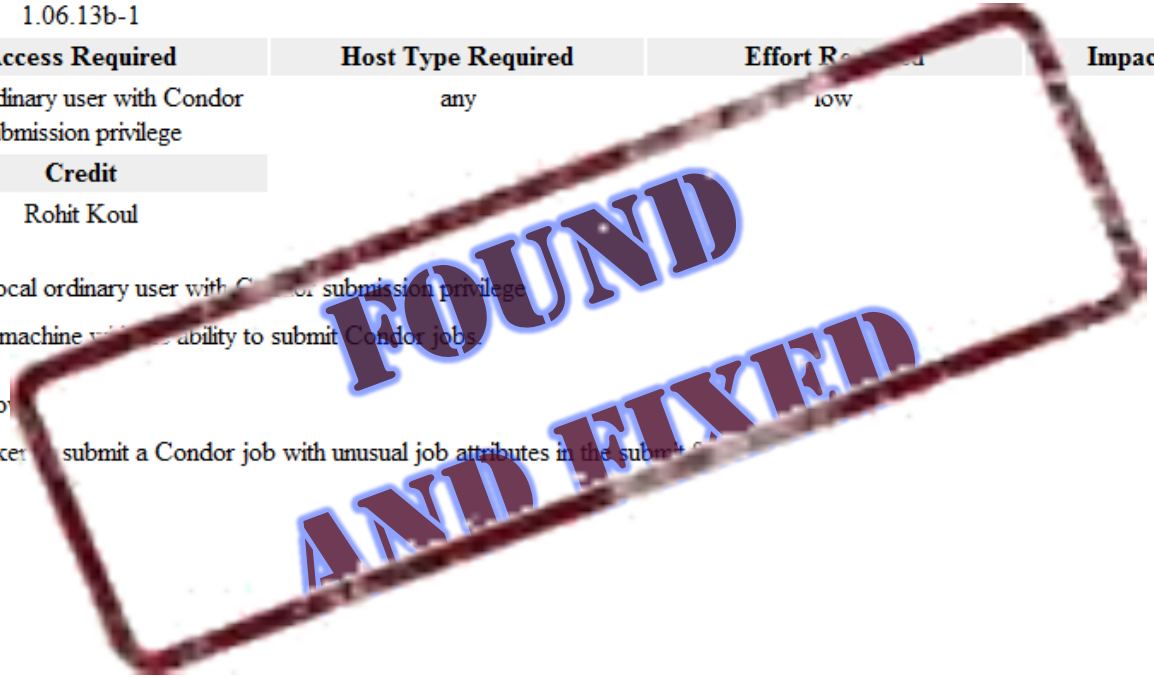
Component	Vulnerable Versions	Platform	Availability	Fix Available
condor_meter	1.04.4d-1 1.06.13b-1	all	not known to be publicly available	no
Status	**Access Required**	**Host Type Required**	**Effort R...**	**Impact/Consequences**
Verified	local ordinary user with Condor submission privilege	any	low	high
Fixed Date	**Credit**			
	Rohit Koul			

Access Required: local ordinary user with C... submission privilege

The vulnerability requires local access to the machine ... ability to submit Condor jobs.

Effort Required: lo...

Exploiting this vulnerability requires the attacker ... submit a Condor job with unusual job attributes in the submit ...



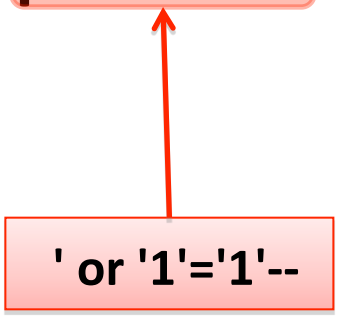
What the bad guys can do

- Gain root access
- Privilege escalation
  - Gain higher privilege access (admin, condor)
- Hijack machines
  - Attack the process running there

What the bad guys can do

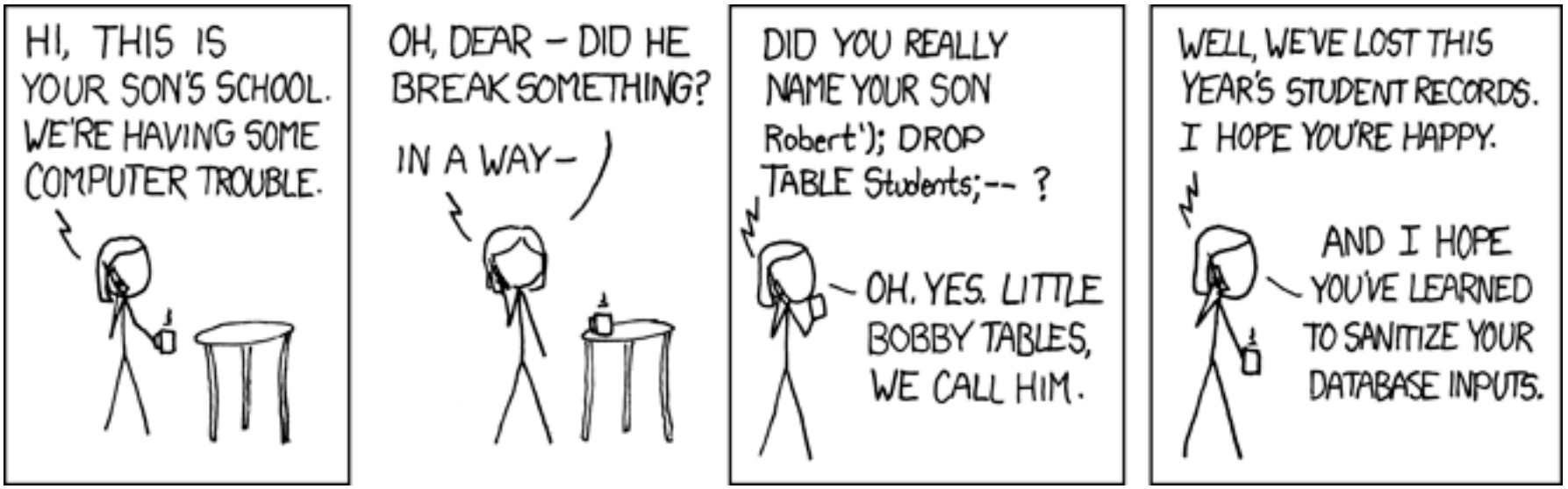
- Injections
  - Command
  - SQL

1. String user = request.getParameter("user");
2. String password = request.getParameter ("password");
3. String sql = "select * from user where username=' " + user + " ' and password=' " + password + " ' ";



What the bad guys can do

- Injections
  - Command
  - SQL



What the bad guys can do

- Injections
  - Command
  - SQL
  - Directory traversal
  - Log
- Denial of Service (DoS)

Why do we care

- Machines belonging to a cloud/grid site are accessible from the Internet
- Hundred of thousands of machines are appealing
- Those machines are continuously probed:
  - Attackers trying to brute-force passwords
  - Attackers trying to break Web applications
  - Attackers trying to break into servers and obtain administrator rights

Why do we do it

- SW has vulnerabilities
- Cloud and Grid SW is complex and large
- Vulnerabilities can be exploited by legal users or by others

Why do we do it

- Attacker chooses the time, place, method, …
- Defender needs to protect against all possible attacks (currently known, and those yet to be discovered)

Key Issues for Security

- Need independent assessment
  - Software engineers have long known that testing groups must be independent of development groups
- Need an assessment process that is NOT based solely on known vulnerabilities
  - Such approaches will not find new types and variations of attacks

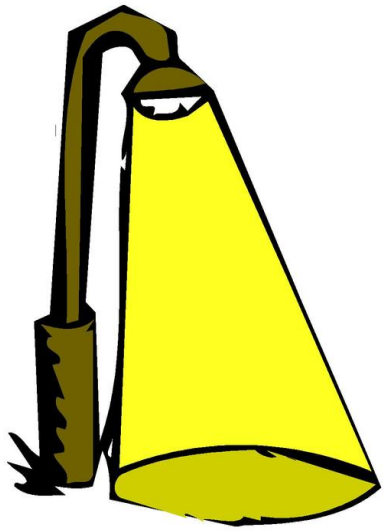
Our Piece of the Solution Space

First Principles Vulnerability Assessment:

- An analyst-centric (manual) assessment process.
- You can't look carefully at every line of code so:

Don't start with known threats …

… instead, identify high value assets in the code and work outward to derive threats.



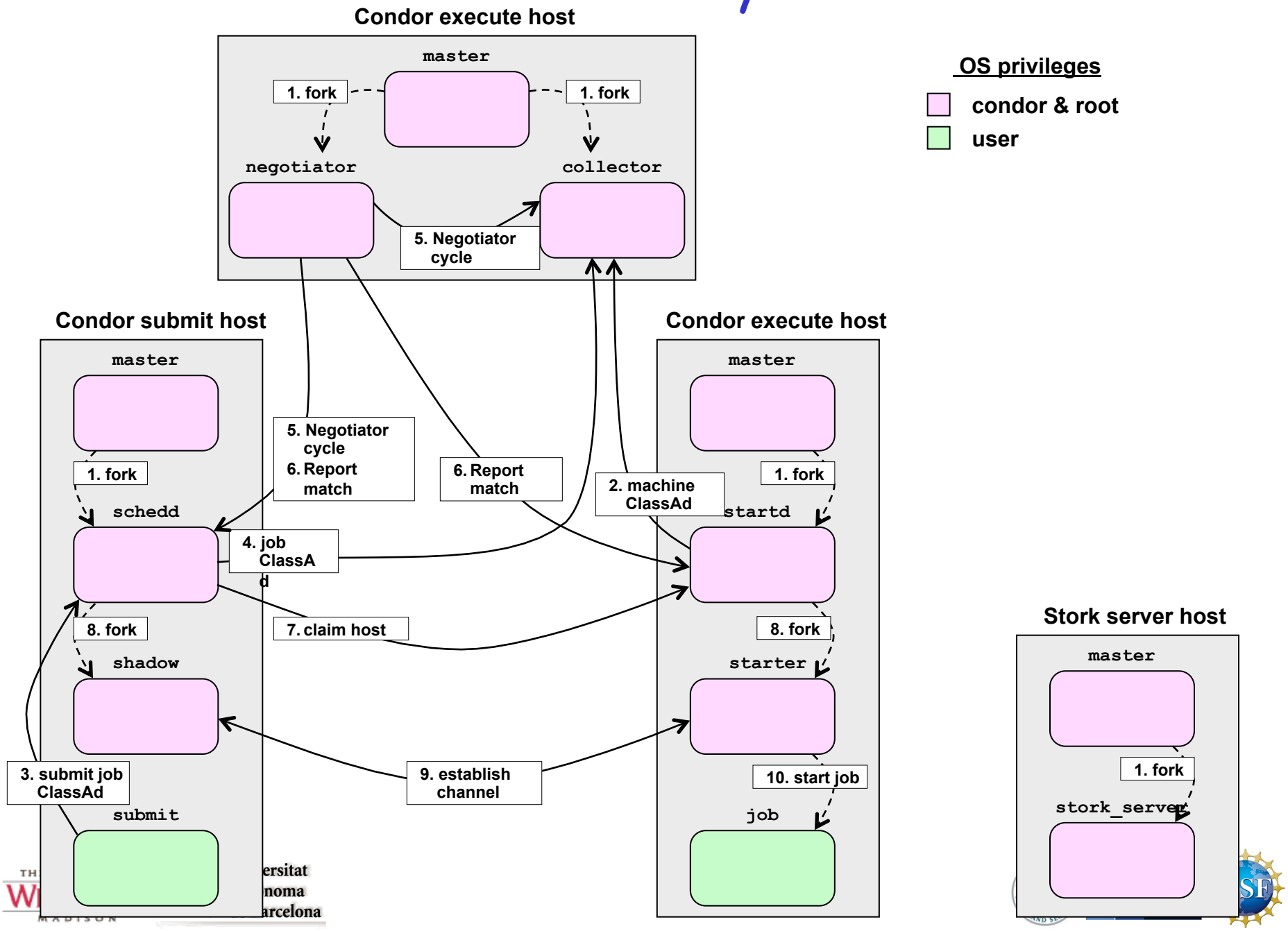
- Start with architectural analysis, then identify key resources and privilege levels, component interactions and trust delegation, then focused component analysis.

First Principles Vulnerability Assessment
Understanding the System

Step 1: Architectural Analysis

- Functionality and structure of the system, major components (modules, threads, processes), communication channels
- Interactions among components and with users

Architectural Analysis: Condor



First Principles Vulnerability Assessment
Understanding the System

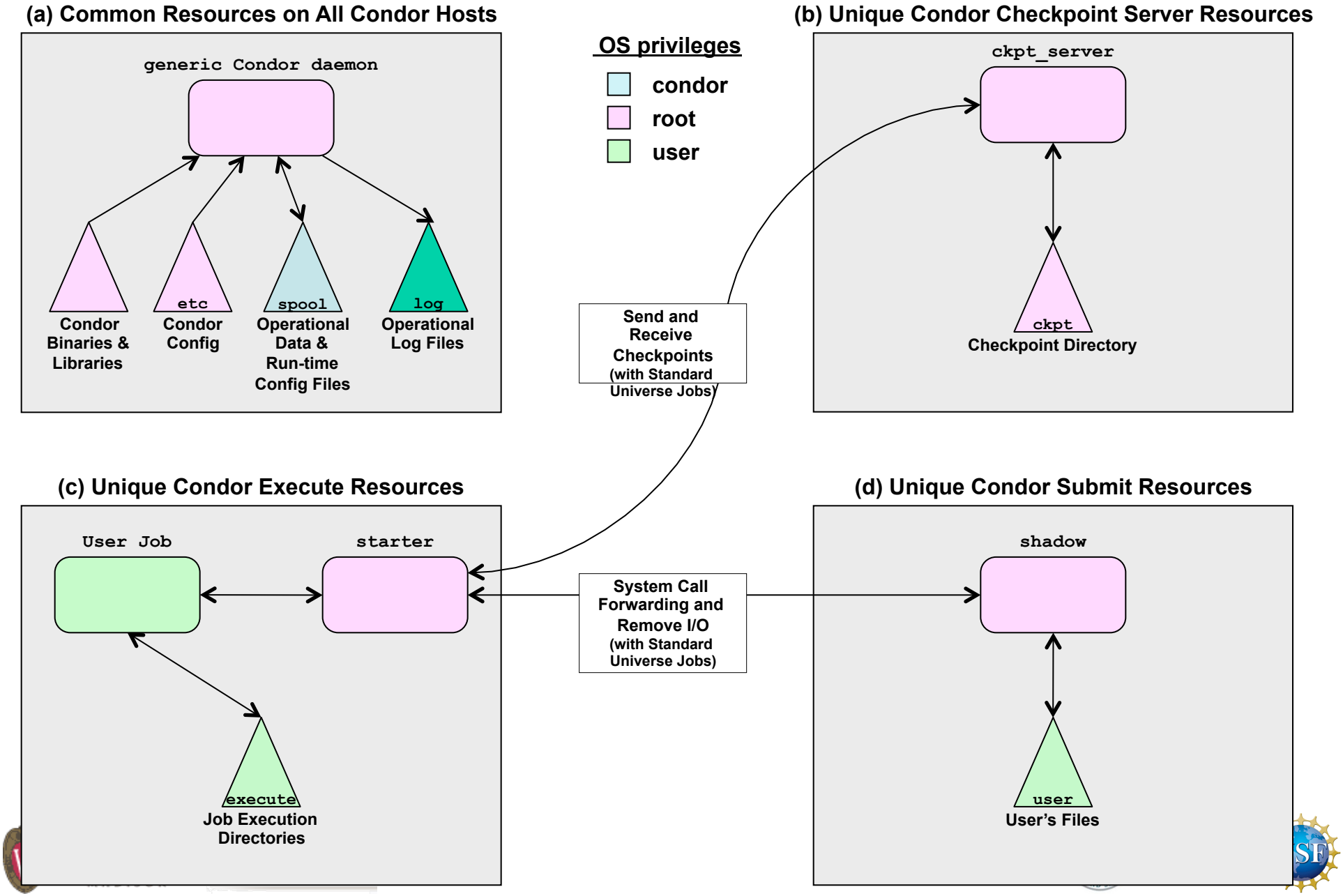
Step 2: Resource Identification

- Key resources accessed by each component
- Operations allowed on those resources

Step 3: Trust & Privilege Analysis

- How components are protected and who can access them
- Privilege level at which each component runs
- Trust delegation

Resource Analysis: Condor



First Principles Vulnerability Assessment
Search for Vulnerabilities

Step 4: Component Evaluation

- Examine critical components in depth
- Guide search using:
  - Diagrams from steps 1-3
  - Knowledge of vulnerabilities
- Helped by Automated scanning tools

First Principles Vulnerability Assessment
Taking Actions

Step 5: Dissemination of Results

- Report vulnerabilities
- Interaction with developers
- Disclosure of vulnerabilities

Our Experience

Condor, University of Wisconsin
Batch queuing workload management system
15 vulnerabilities — 600 KLOC of C and C++

SRB, SDSC
Storage Resource Broker - data grid
5 vulnerabilities — 280 KLOC of C

MyProxy, NCSA
Credential Management System
5 vulnerabilities — 25 KLOC of C

glExec, Nikhef
Identity mapping service
5 vulnerabilities — 48 KLOC of C

Gratia Condor Probe, FNAL and Open Science Grid
Feeds Condor Usage into Gratia Accounting System
3 vulnerabilities — 1.7 KLOC of Perl and Bash

Condor Quill, University of Wisconsin
DBMS Storage of Condor Operational and Historical Data
6 vulnerabilities — 7.9 KLOC of C and C++

Our Experience

Wireshark, wireshark.org
Network Protocol Analyzer
in progress 2400 KLOC of C

Condor Privilege Separation, Univ. of Wisconsin
Restricted Identity Switching Module
21 KLOC of C and C++

VOMS Admin, INFN
Web management interface to VOMS data
35 KLOC of Java and PHP

CrossBroker, Universitat Autònoma de Barcelona
Resource Mgr for Parallel & Interactive Applications
97 KLOC of C++

Our Experience

ARGUS 1.2, HIP, INFN, NIKHEF, SWITCH
gLite Authorization Service
in progress

NIKHEF

glExec 0.8, Nikhef
Identity mapping service

What do we do

- Make cloud/grid software more secure
- Make in-depth assessments more automated
- Teach tutorials for users, developers, admin, managers:
  - Security risks
  - Vulnerability assessment
  - Secure programming

Who we are

Bart Miller
Jim Kupsch
Karl Mazurak
Rohit Koul
Daniel Crowell
Wenbin Fang

Elisa Heymann
Eduardo Cesar
Jairo Serrano
Guifré Ruiz
Manuel Brugnoli

Security Risks in Clouds and Grids

Elisa Heymann

Elisa.Heymann@uab.es

Barton P. Miller
James A. Kupsch

bart@cs.wisc.edu

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

http://www.cs.wisc.edu/mist/papers/VAshort.pdf