Universitat Autònoma de Barcelona

First Principles Vulnerability Assessment

Elisa Heymann
Manuel Brugnoli

Computer Architecture &
Operating Systems Department
Universitat Autònoma de Barcelona

The Bad News

- The bad guys are trying to do *really bad* things to us.
- They are smart, dedicated and persistent.
- No single approach to security can be sufficient.
- The attackers have a natural advantage over the defenders.

We have to approach the defense of our systems as *security in depth*.

The Good News

- 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

- We found some serious vulnerabilities … and more vulnerabilities … and more.

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

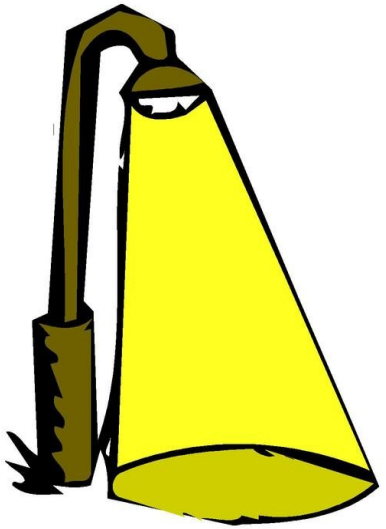
Our Piece of the Solution Space

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.



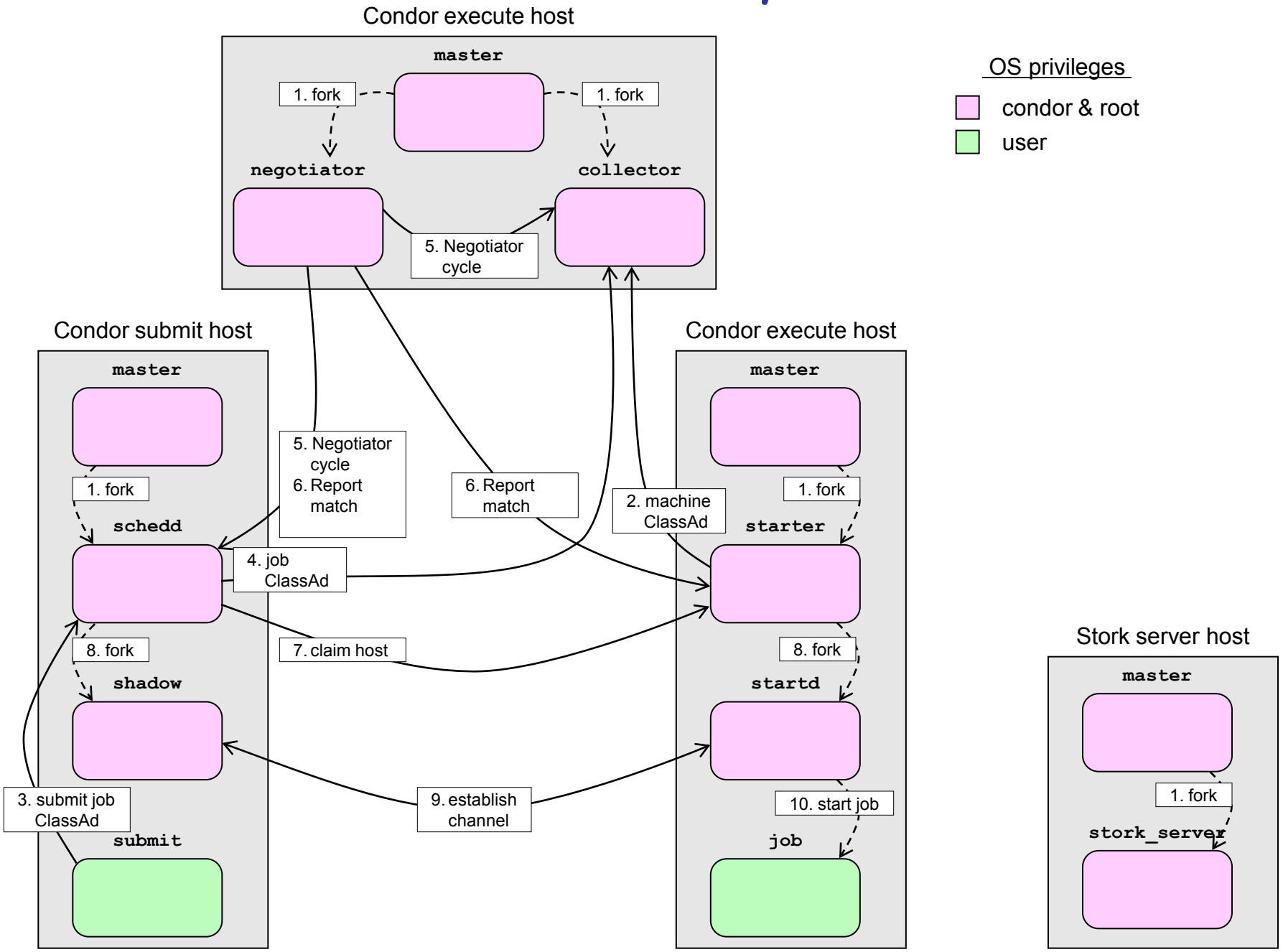
- Start with architectural analysis, then identify key resources and privilege levels, component interactions and trust delegation, then focused component analysis.

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

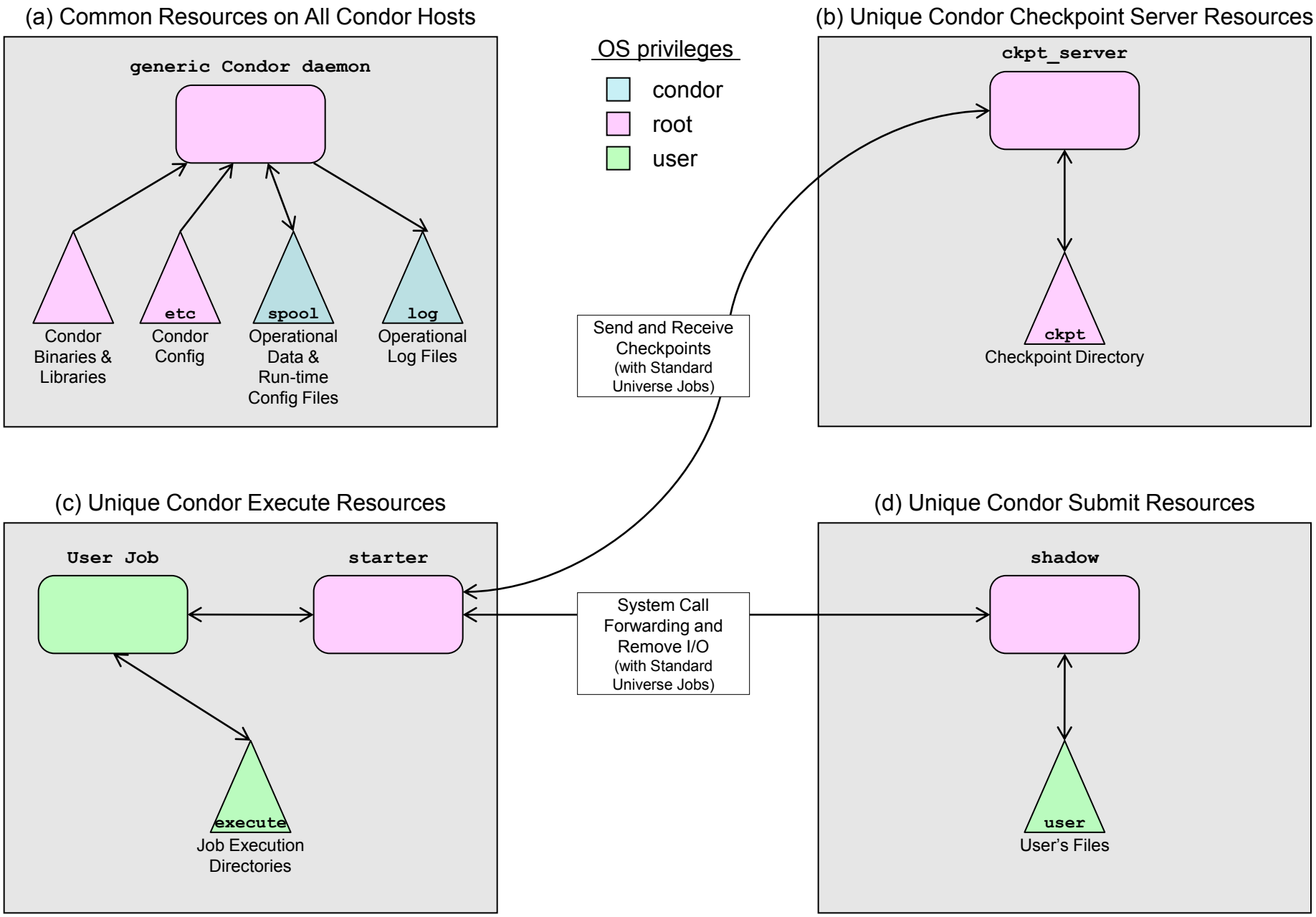
Step 2: Resource Identification

- 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

First Principles Vulnerability Assessment Taking Actions

THE UNIVERSITY of WISCONSIN MADISON

CONDOR-2005-0003

SDSC

Summary:

Arbitrary commands can be executed with the permissions of the condor_shadow or condor_gridmanager's effective uid (normally the "condor" user). This can result in a compromise of the condor configuration files, log files, and other files owned by the "condor" user. This may also aid in attacks on other accounts.

Component	Vulnerable Versions	Platform	Availability	Fix Available
condor_shadow condor_gridmanager	6.6 - 6.6.10 6.7 - 6.7.17	all	not known to be publicly available	6.6.11 - 6.7.18 -
Status	**Access Required**	**Host Type Required**	**Effort Required**	**Impact/Consequences**
Verified	local ordinary user with a Condor authorization	submission host	low	high
Fixed Date	**Credit**			
2006-Mar-27	Jim Kupsch			

Access Required: local ordinary user with a Condor authorization

This vulnerability requires local access on a machine that is running a condor_schedd, to which the user can use condor_submit to submit a job.

Effort Required: low

To exploit this vulnerability requires only the submission of a Condor job with an invalid entry.

Impact/Consequences: high

Usually the condor_shadow and condor_gridmanager are configured to run as the "condor" user, and this vulnerability allows an attacker to execute arbitrary code as the "condor" user.

Depending on the configuration, additional more serious attacks may be possible. If the configuration files for the condor_master are writable by condor and the condor_master is run with root privileges, then root access can be gained. If the condor binaries are owned by the "condor" user, these executables could be replaced and when restarted, arbitrary code could be executed as the "condor" user. This would also allow root access as most condor daemons are started with an effective uid of root.

Studied Systems

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

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++

Studied Systems

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 (role

35 KLOC of Java and PHP

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

In Progress for EMI

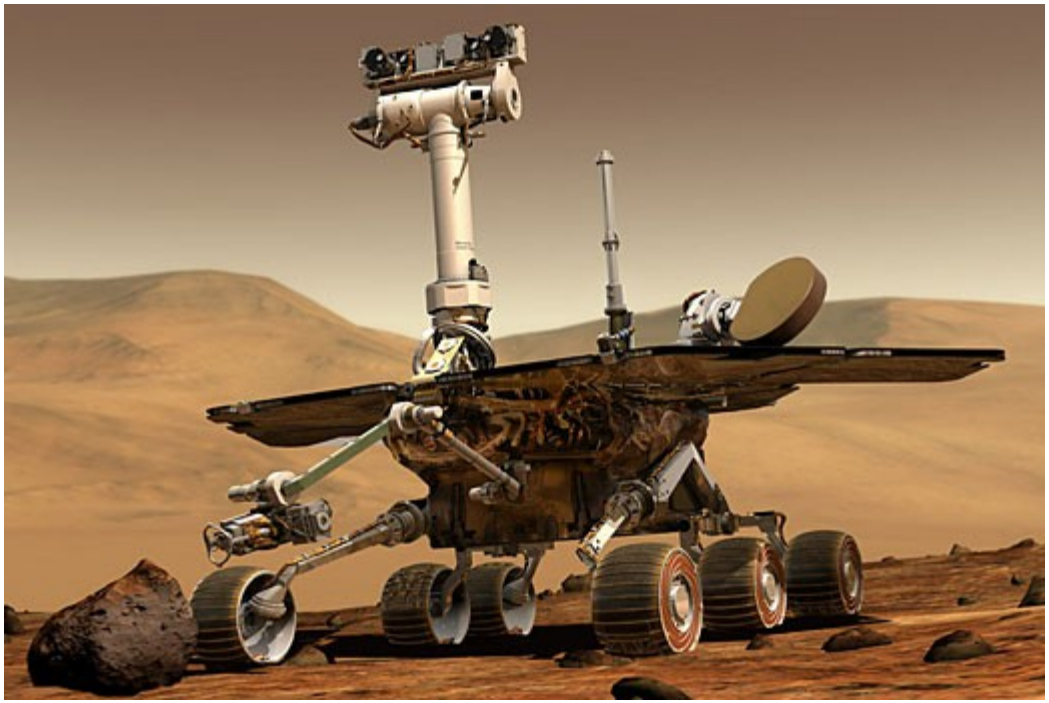
ARGUS, wireshark.org
gLite Authorization Service
in progress

glExec 0.8, Nikhef
Identity mapping service
in progress

What about Automated TOOLS?

- Everyone asks for them
- They may help but …

… they are definitely not enough!



Manual vs. Automated Vulnerability Assessment

The literature on static analysis tools is self-limiting:

- Missing comparison against a ground truth
- Tool writers write about what they have found
- Limited discussion of false positives

Every valid new problem that a tool find is progress, but it's easy to lose perspective on what these tools are *not* able to do

EMI

- Roadmap needed:
  - gridFTP
  - CREAM
  - WMS
- We need input from you!

How do You Respond?

A change of culture within the development team:

- When security becomes a first-class task, and when reports start arriving, *awareness* is significantly increased.
- This effects the way developers look at code *and* the way that they write code.
- A major landmark: when your developers start reporting vulnerabilities that they've found on their own.

Thank you.

Elisa.Heymann@uab.es