Vulnerability Assessment for Middleware

Elisa Heymann, Eduardo Cesar
Universitat Autònoma de Barcelona, Spain

Jim Kupsch, Barton Miller
University of Wisconsin-Madison

Barcelona, September 21st 2009

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 on known vulnerabilities
  - Such approaches will not find new types and variations of attacks

Key Issues for Security

- Automated Analysis Tools have Serious Limitations
  - While they help find some local errors, they
    - MISS significant vulnerabilities (false negatives)
    - Produce voluminous reports (false positives)
- Programmers must be security-aware
  - Designing for security and the use of secure practices and standards does not guarantee security

Addressing these Issues

- We must evaluate the security of our code
  - The vulnerabilities are there and we want to find them first
- Assessment isn't cheap
  - Automated tools create an illusion of security
- You can't take shortcuts
  - Even if the development team is good at testing, they can't do an effective assessment of their own code

Addressing these Issues

- Try First Principles Vulnerability Assessment
  - A strategy that focuses on critical resources
  - A strategy that is not based on known vulnerabilities
- We need to integrate assessment and remediation into the software development process
  - We have to be prepared to respond to the vulnerabilities we find

Talk Agenda

- First Principles Vulnerability Assessment: a new strategy
- Results of applying FPVA
- A concrete evaluation of automated assessment tools
- Some follow-up discussion

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

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

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
Step 5: Dissemination of Results

CONDOR-2005-0003

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.

FPVA - Studied Systems

Condor
- University of Wisconsin
- Batch queuing workload management system

SRB
- SDSC
- Storage Resource Broker - data grid

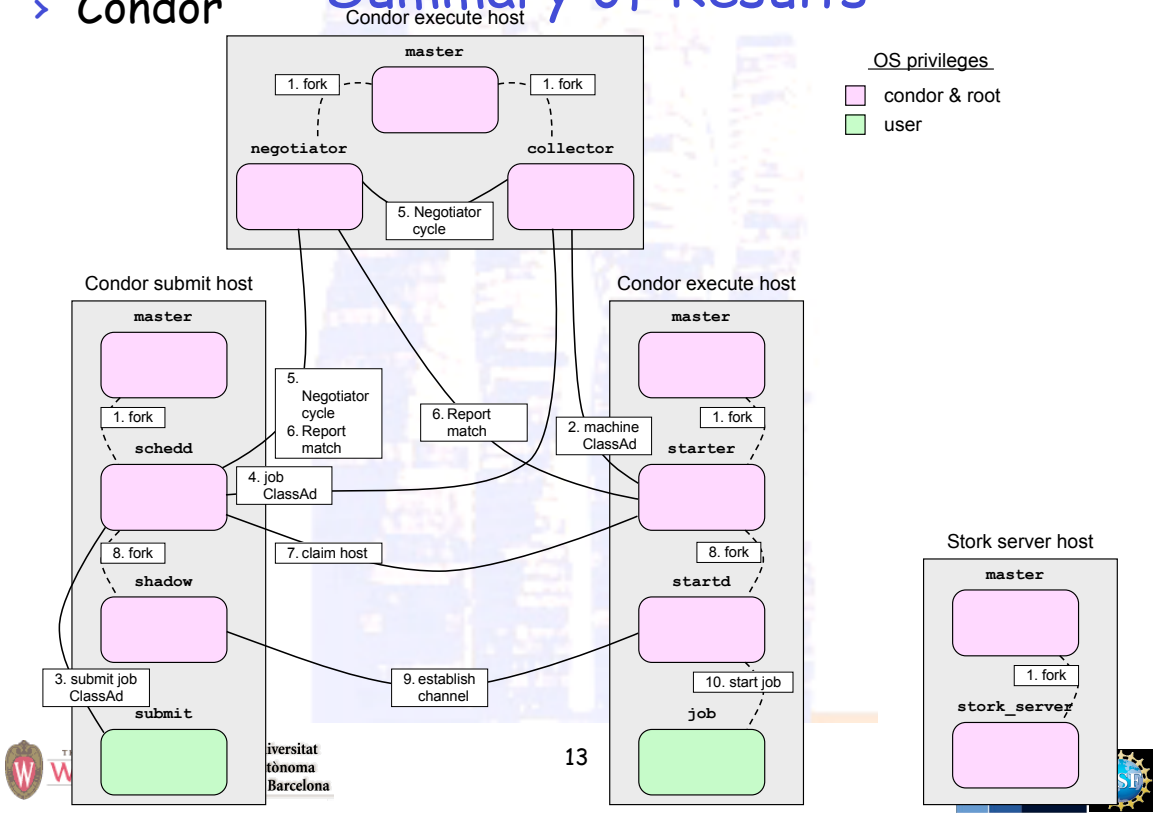
MyProxy
- NCSA
- Credential Management System

glExec (in progress)
- NIKHEF
- Identity mapping service

CrossBroker (in progress)
- Universitat Autònoma de Barcelona
- Resource Manager for Parallel and Interactive Applications

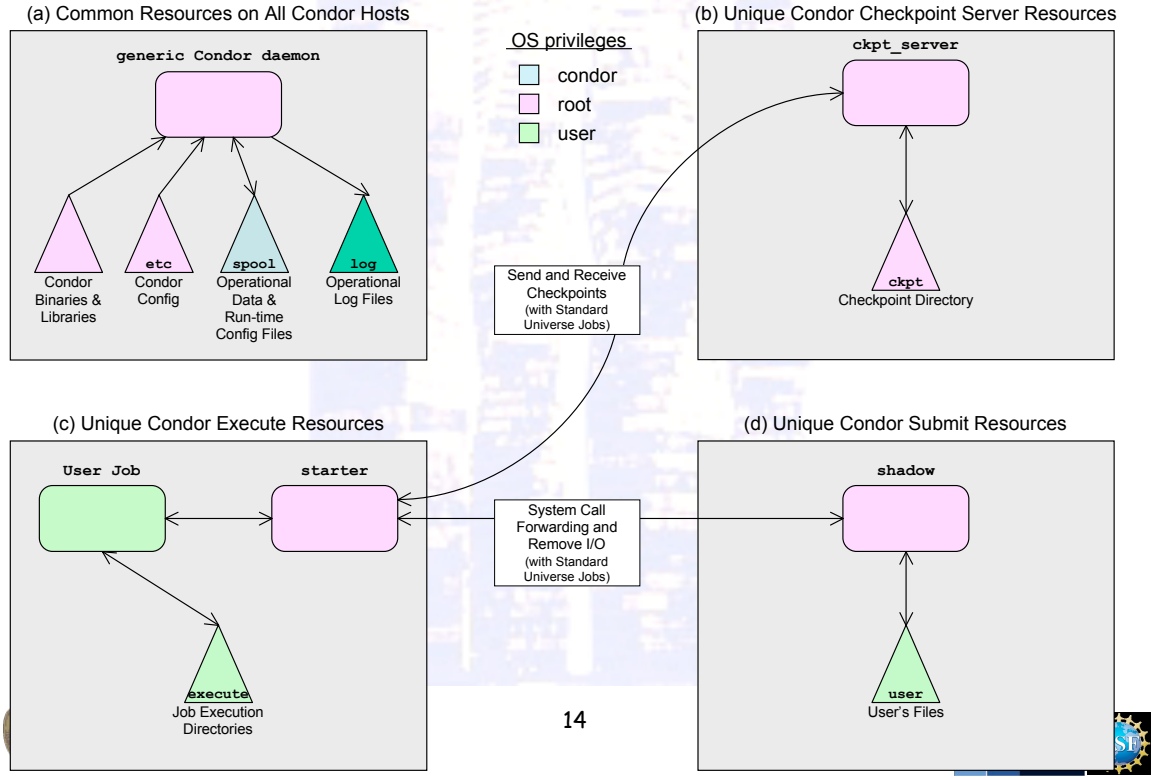
Summary of Results

- Condor



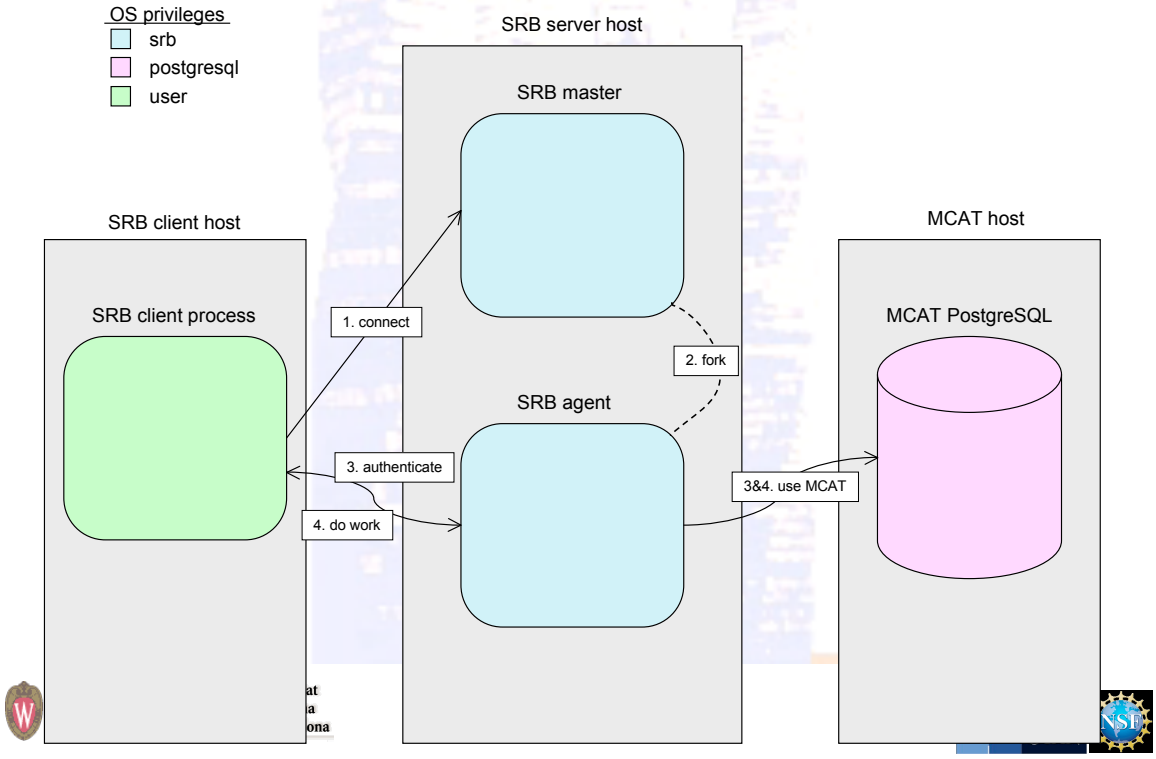
Summary of Results

- Condor



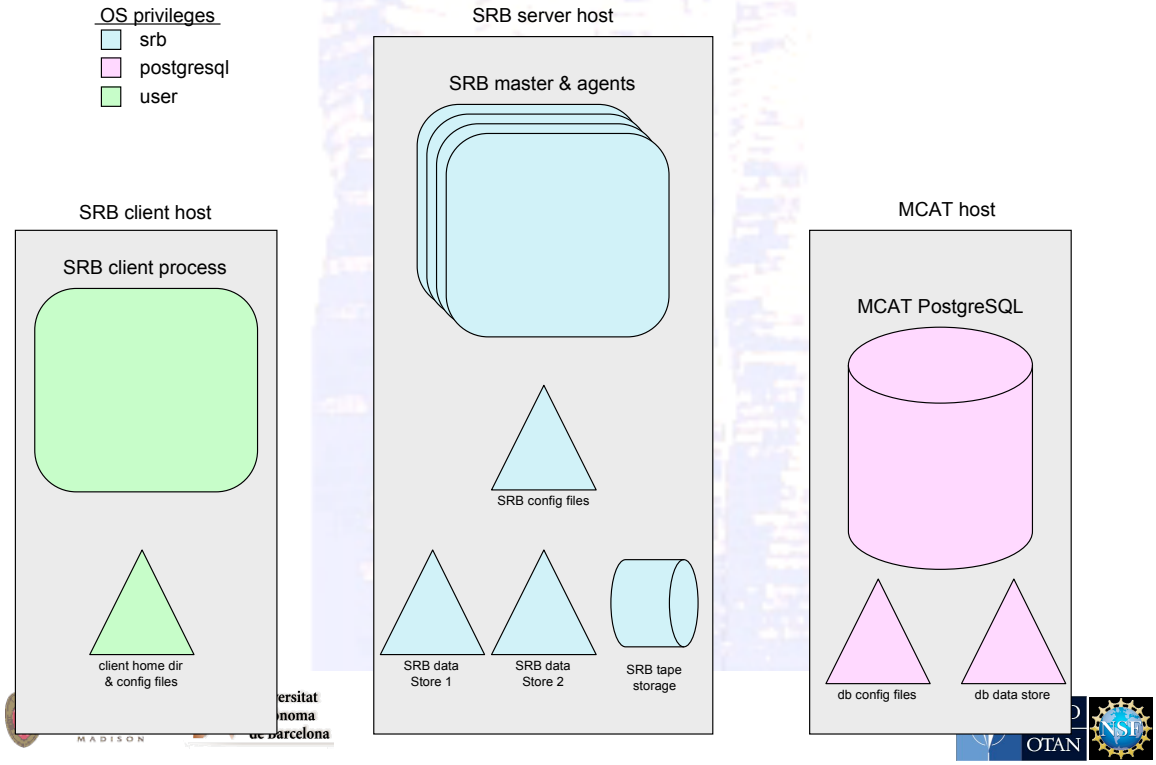
Summary of Results

- SRB



Summary of Results

- SRB



Summary of Results

- Examples of problems found:
  - Directory traversal allows reading/writing any file
  - Authentication problems
  - Command injections allow arbitrary code to be executed
  - Simple programming errors trigger gaining root access to the system
  - Denial of services
  - …

Summary of Results
First Principles Vulnerability Assessment

Technique has been extremely successful

- found critical problems
- helped groups redesign software
- changed their development practices and release cycle management

What about Automated TOOLS?

- Everyone asks for them
- They may help but …

They are not enough!

Manual vs. Automated Vulnerability Assessment

- Literature on static analysis tools, papers are almost self limiting:
  - missing comparison against security as a whole
  - tool writers write about what they have found
- Every valid new thing tools find is progress, but it's easy to lose perspective on what these tools are not able to do

Case Study: Methodology

- Assessed Condor using FPVA
- Identified the best Automated Tools
- Applied these tools to the same version of Condor as was used in the FPVA study
- Goal: to compare the ability of these tools to find serious vulnerabilities (having a low false negative rate), while not reporting a significant number of false vulnerabilities or vulnerabilities with limited exploit value (having a low false positive rate)

Manual Assessment: 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

Case Study

Goal is to study the best tools out there
Apply them to the same system we studied

- Talked to academics, military, and industry people about what they thought were the best tools:
  - Coverity Prevent 4.1.0
  - Fortify SCA 5.1.0016
- Review tool output

Tool Result Summary

	Coverity	Fortify	
Defects Found:	2,986	15,466	total
		3	critical
		2,301	hot
		8,101	warm
		5,061	info
Manual Defects Found:	1	6	total
	1	6	impl. bug
	0	0	design flaw

Manual Tool Discovered Defects

- Simple implementation bugs found
  - Coverity found 1
    - errors on the side of false negatives
    - only flags certain functions when input can be proven to come from untrusted sources
  - Fortify found 6
    - errors on the side of false positives
    - will always flag certain functions

- No design flaw defects found

Manual Tool Comparison Study

- Showed limitations of current tools
- Presented manual vulnerability assessment as a required part of a comprehensive security assessment
- Created a reference set of vulnerabilities to perform apples-to-apples comparisons

Our Work -- Summary

Assess: We continue to assess new software systems

Train: We present tutorials and white papers, and continue to develop new educational materials

Research: Our results provide the foundation for new research to make FPVA less labor-intensive and improve quality of automated code analysis

Summary

The bad news: No easy or cheap solution.

So, what should you do:

- Programmers: Learn secure programming
- Managers: Prioritize security, invest in it, and have assessment and response strategies.

- Come talk to us
  - Advice
  - Tutorials
  - Assessment

Elisa.Heymann@uab.es

VA Tutorial

When? Friday Sept. 25th 8:30-10:00
& 11:00-13:00

Where? Montjuic room

More information on FPVA:
http://www.cs.wisc.edu/mist/VA.pdf
Elisa.Heymann@uab.es

Questions?