Distributed password cracking with Condor and John the Ripper

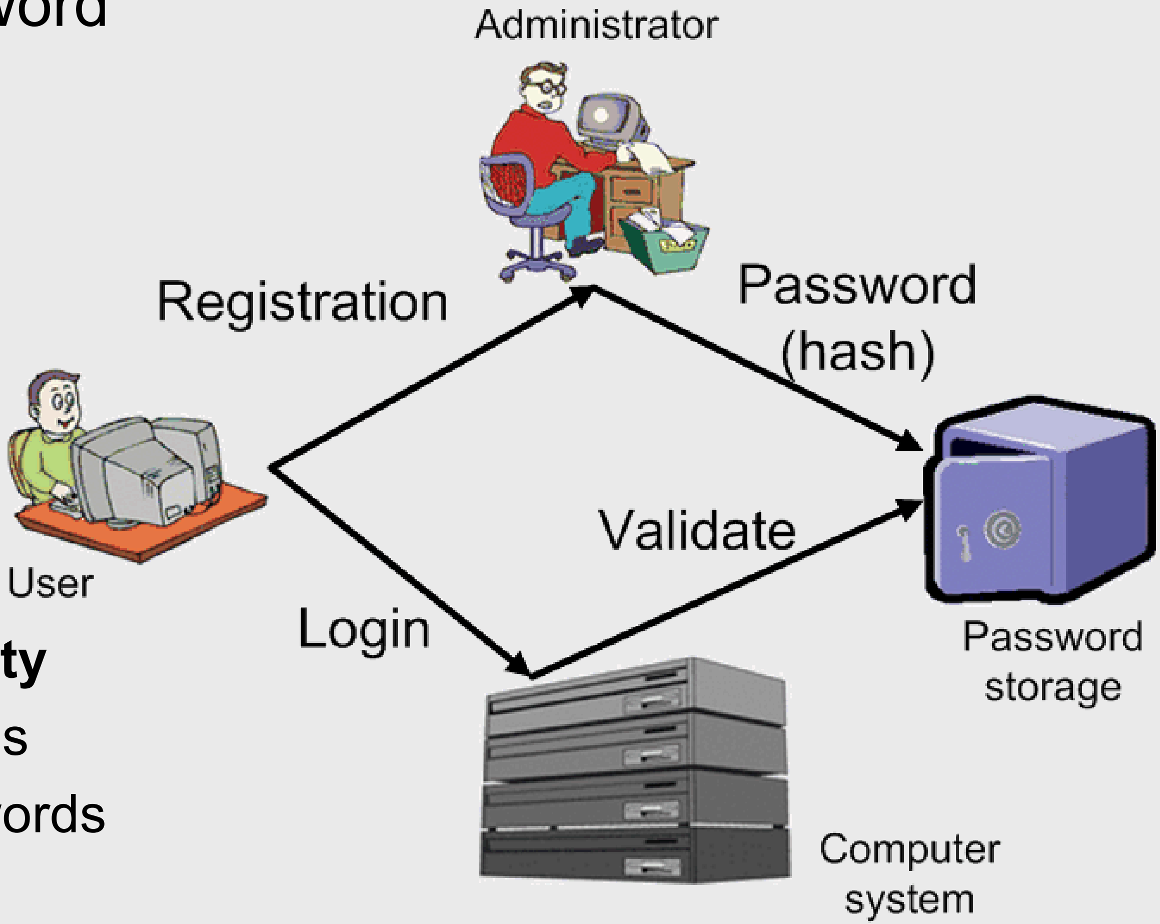
E. Imamagic, A. Dmitrovic, S. Buljat

Overview

- Introduction
  - Username/password authentication
- Password Cracking
  - John the Ripper
- Condor
- Our approach
- Conclusion
- Future work

Introduction

- Authentication
- Username/password authentication
- Hash:
  - MD5
  - SHA-1
  - Blowfish
  - …
- Issues
  - **Password quality**
  - Stolen passwords
  - Forgotten passwords

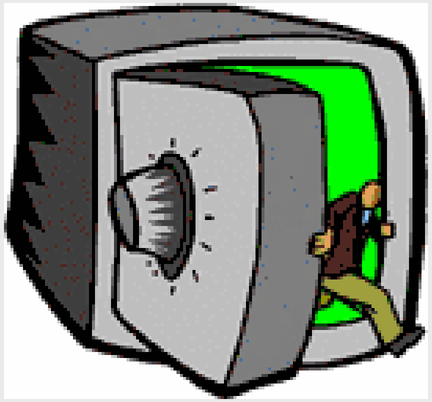


Password cracking

- Recovering the password from safe storage
- Algorithm:

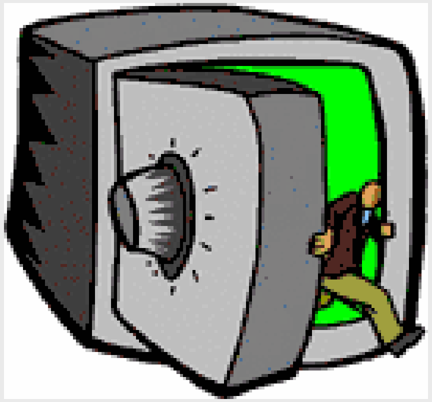
```
HashedPass=GetPassHash(userDb);
repeat{
  Guess=GeneratePasswordGuess();
  HashedGuess=Hash(Guess);
  passFound=Compare(HashedGuess,HashedPass);
}until(passFound);
```

- Password generation approaches:
  - Guessing
  - Dictionary attack
  - Brute force



John The Ripper

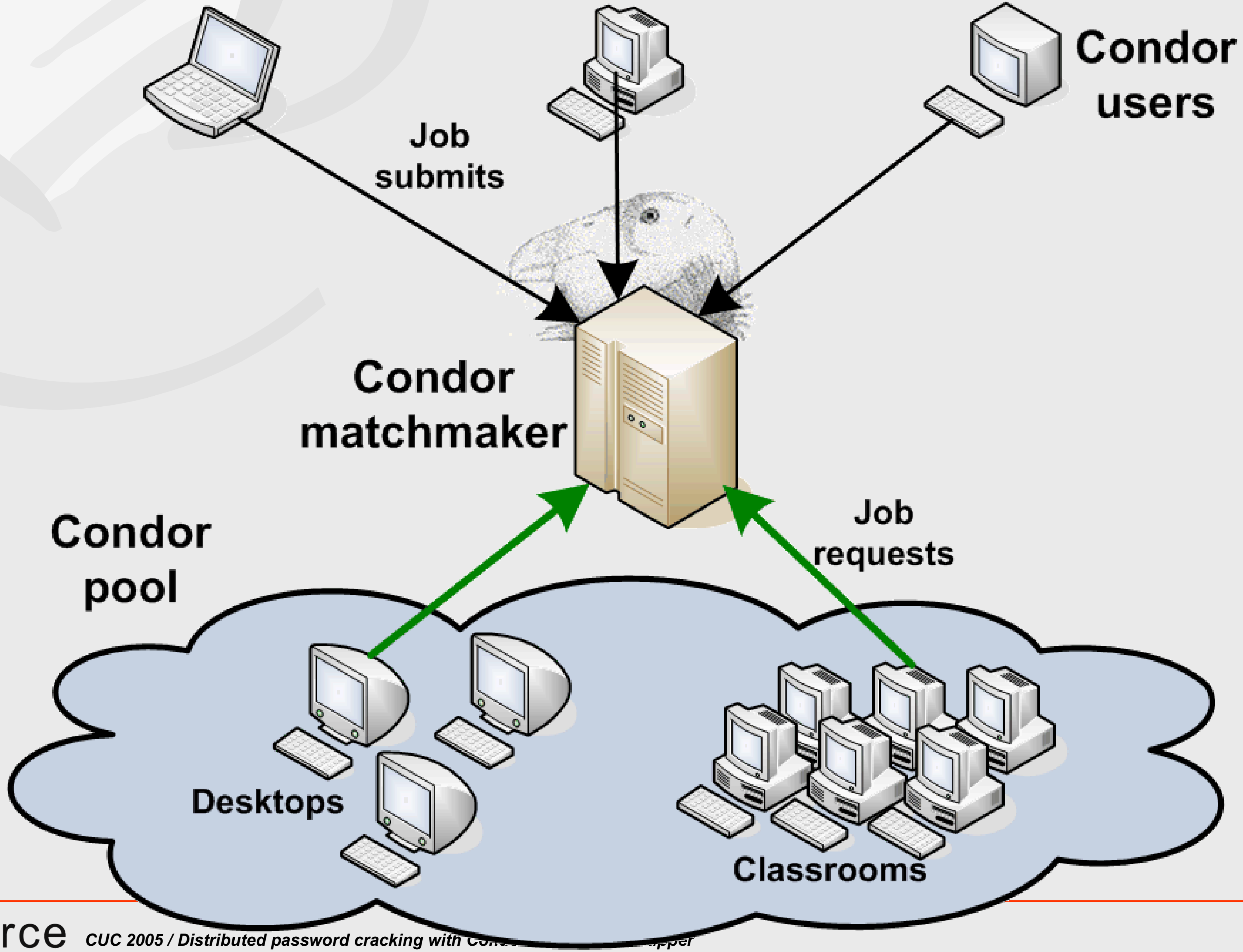
- Open source cracking tool
- Optimized speed & memory usage
- Enables all three approaches
- Creation of custom guessing rules
- **Checkpointing**
- Supports various password storage:
  - UNIX, DOS, MS Windows, OpenVMS, BeOS
  - MySQL, AFS, Apache htpasswd
- Ported to various platforms:
  - UNIX, MS Windows, Dos
- http://www.openwall.com/john/



Condor

- High throughput computing
  - Large arrays of independent tasks
  - Single Instruction Multiple Data (**SIMD**)
- Utilization of available resources
  - CPU Harvesting, SETI@Home
- Main features
  - Support for heterogeneous environments
  - File transfer and remote I/O
  - Integration with grid technologies
  - Complex jobs: workflows & parallel jobs
  - Checkpointing
- http://www.cs.wisc.edu/condor/

Condor


Job submits
Condor users
Condor matchmaker
Job requests
Condor pool
Desktops
Classrooms

Our approach

- John + Condor
- Set of passwords is split in groups
- Each group is submitted to Condor pool
- Condor finds available computer and executes John
- If computer becomes occupied, Condor
  - packs all needed data
  - migrates John on another computer

Results

- Equipment:
  - 7 * Fujitsu Pentium III, 930 MHz, 256 & 512 MB – **Linux**
  - 2 * desktops – **MS Windows**
- Currently we've been running John for **136** days
- Passwords broken:
  - 10% (20 of 200)

Conclusion & future work

- Password cracking
  - ideal high throughput application
  - John can be migrated
- Optimizing Condor + John integration
- Investigating approaches for passphrase cracking
  - Optimization of dictionary attack
- Utilizing Condor for other purposes
  - Image processing