Automatically Creating Realistic Targets for Digital Forensics Investigation

By

Frank Adelstein, Yun Gao and Golden Richard

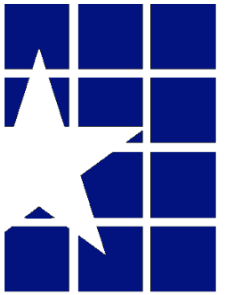
Presented At

The Digital Forensic Research Conference

DFRWS 2005 USA New Orleans, LA (Aug 17th - 19th)

DFRWS is dedicated to the sharing of knowledge and ideas about digital forensics research. Ever since it organized the first open workshop devoted to digital forensics in 2001, DFRWS continues to bring academics and practitioners together in an informal environment. As a non-profit, volunteer organization, DFRWS sponsors technical working groups, annual conferences and challenges to help drive the direction of research and development.

http:/dfrws.org



Automatically Creating Realistic Targets for Digital Forensic Investigation

Dr. Frank Adelstein, ATC-NY

Yun Gao, Prof. Golden G. Richard, III
University of New Orleans

Digital Forensic Research Workshop
New Orleans, LA
August 17, 2005

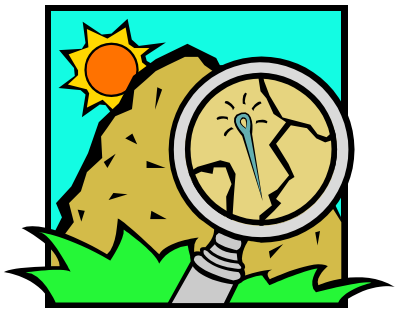
Overview

- Introduction
  - Challenges of teaching forensics
  - Forensics tools - OnLine Digital Forensic Suite™
- Classroom Experiment
  - Results
  - Lessons Learned
- Need for automation
- Summary

Introduction

- The volume of digital evidence continues to grow
  - criminal and civil cases (e-discovery), incident reponse, etc.
- The need for highly trained digital forensic investigators is also increasing
- More and more colleges are offering computer forensic courses
- Good hands-on training is essential to supplement traditional training
- “Good” training *must* be realistic
- Creating a good lab is tedious

What is a realistic lab?



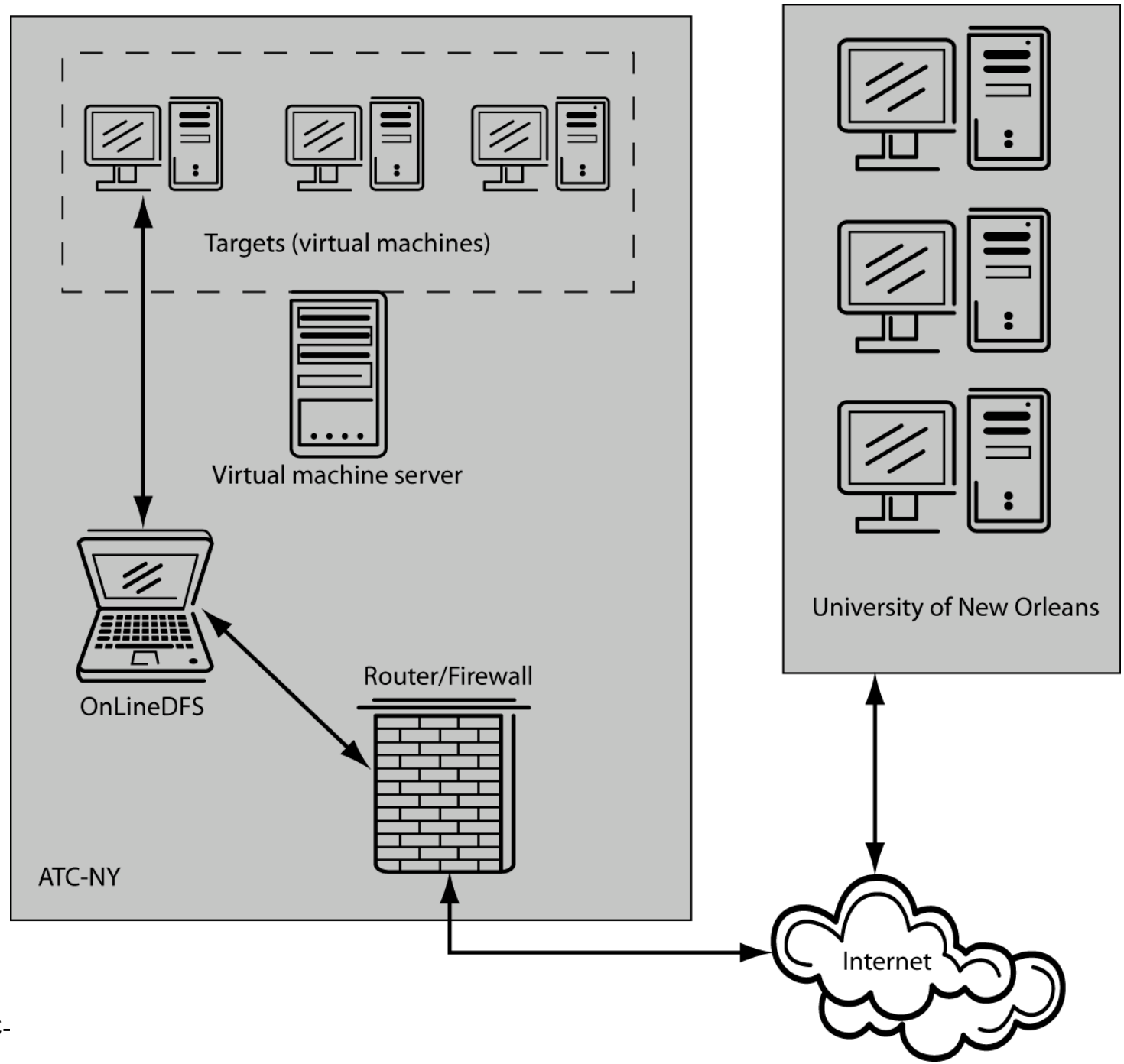
- Analogy: needle in a haystack
  - Floppy disk: small haystack
  - 40 G disk: big haystack
- Lots of “stuff” happening on a real system
  - Processes, files, deleted files, log entries, timestamps, history of things that happened, connections among them
  - Lots of useless, irrelevant stuff on a system, dead-ends
  - Can’t just scan through with a disk editor

Tools

- Mobile Forensic Platform (DFRWS 2002) developed to allow investigators to perform analysis of a live system remotely.
    - Now available as OnLine Digital Forensic Suite™ and LiveWire Investigator ™.
- We wanted to see if we could create a realistic lab for students to investigate live systems remotely using our tools.
- Used UNO computer forensic class to test it and ATC-NY lab facilities in Ithaca

Classroom Experiment

- Students investigate a single target machine on a network
- Network hidden behind firewall
- OnLineDFS machine behind firewall, only HTML traffic to OnLineDFS permitted inbound through firewall.
- Target machine is actually a virtual machine
  - 2 VMware servers running 3 VMs each
  - 6 student groups
- Created a “simple” scenario


Targets (virtual machines)
Virtual machine server
OnLineDFS
Router/Firewall
ATC-NY
University of New Orleans
Internet

ATC-

Assignment

- Called in to investigate problems of misuse of computer resources at a company.
- Determine who, where, what, why, when, how, …
- "Penguin pornography" – any depiction of the birds *without* hats.
  - Whimsical "crime" with real world analogy

- (Full text of assignment is in the following slide.)

Assignment

Introduction: In the FreeeZ-E-Q Ice Cube Company, the owner of the machines you will investigate, there has been a rising problem with employees accessing illegal penguin porn. Legally, penguin porn includes any depictions of penguins not wearing hats (the top of a penguin's head is defined as the "dirty bits"). You are to investigate the target machine and answer questions in the following list. You can assume that any "bad guys" are very technically savvy and have employed many tricks to conceal their activities.

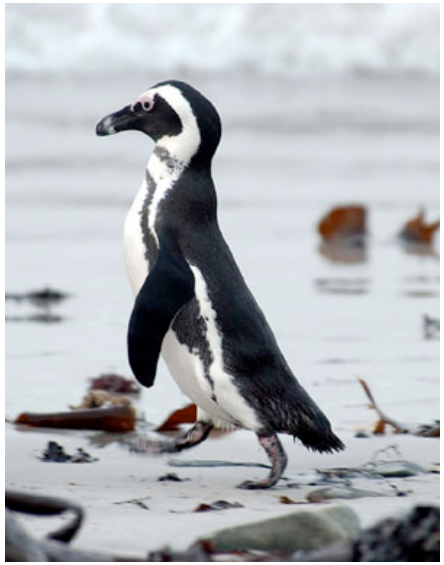
1. Who's the bad guy?
2. Is there penguin porn on the box?
3. If there's penguin porn, how is it being distributed?
4. What clever techniques, if any, were used to obscure the activities?
5. (Exhaustively) where on the box is the bad guy storing applications, data associated with their dark, evil crime, or the penguin porn itself?
6. Has the bad guy attempted to implicate anyone else who uses the target machine in the crime? If so, who?
7. What is the numeric IP address of the site of the bad guy's supplier?
8. Are there any password-protected pages on the supplier's site? If so, provide URLs, usernames and passwords.
9. What is the exact hostname of the site in question 7? Hint: virtual hosting is probably used, so a simple reverse lookup will not give you the correct answer.
10. What is the name of the supplier of penguin porn? Hint: Not the bad guy mentioned in Q1!
11. What species of penguin does the supplier not have pictures of? Does he say anywhere when he might have this type again?

The Evidence

- Wholesome



- Naughty




Photos courtesy Michael Leibow and Jen Beaven, www.pencognito.com.

Scenario

- 10 users defined (Ann, Bob, Cindy, Dan, …)
- One user, Cindy, is running an illegal web server, (re)named vim, hidden in a software development directory
- DocumentRoot for web server is in another user's directory (due to world-writable permissions)
- "Dirty" (hatless) pictures in directory with 1000+ flower photos
- Cindy is running lynx (text-only web browser), viewing a password protected page from "the supplier" (URL only present in process memory)
- Account info stored in a ~/.source file. Password stored in comment of jpeg file in rot13 format
- .bash_history shows *how* password stored (but not *what*)
- Other users on system compiling and running programs, …

Results

- All students completed assignment
- In debrief, students reported spending 5-20 hours on assignment
- Most student had positive comments about lab (some comments on downtime and speed)
- Most would have liked to see a harder, more complex lab with more dead-ends
- Users not involved in scenario had no .bash_history and little activity – too easy for students to spot

Lessons Learned

- Designing and implementing labs takes a lot of time
- The easiest way to make a proper history trail is to enact the events for real
- Customizing or changing labs (different users, different commands) takes time and is hard to track
- Analyzing OnLineDFS log files is tedious – lots of information to sort through
- Automation tools would help

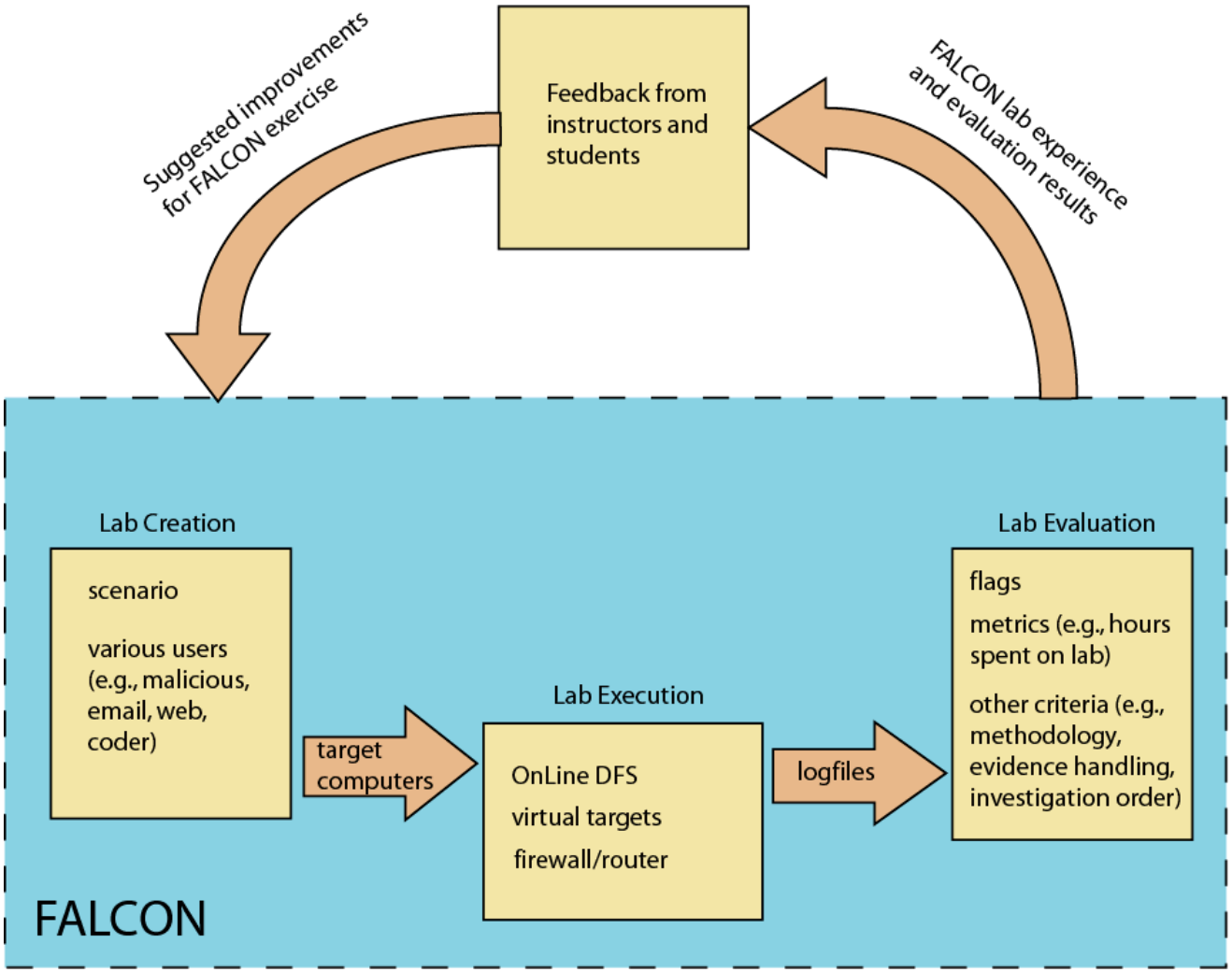
FALCON

Framework for Laboratory Exercises Conducted Over Networks

3 main components:

1. Lab Creation Tool (LCT)
2. Lab Execution Environment (LEE)
3. Lab Evaluation Tool (LET)

FALCON Cycle



FALCON Lab Creation Tool

- Automatically specify machine details based on configurations
- Generate users by roles, using a list of names
- Generate events by executing command sequences (so history and logs will be accurate)
- “Time warp” (i.e., change the system clock) to get proper timestamps
- Can randomize users (e.g., bad guy) and have events be probabilistic with dependencies on previous events, and time relative to prior events

FALCON Lab Creation Tool

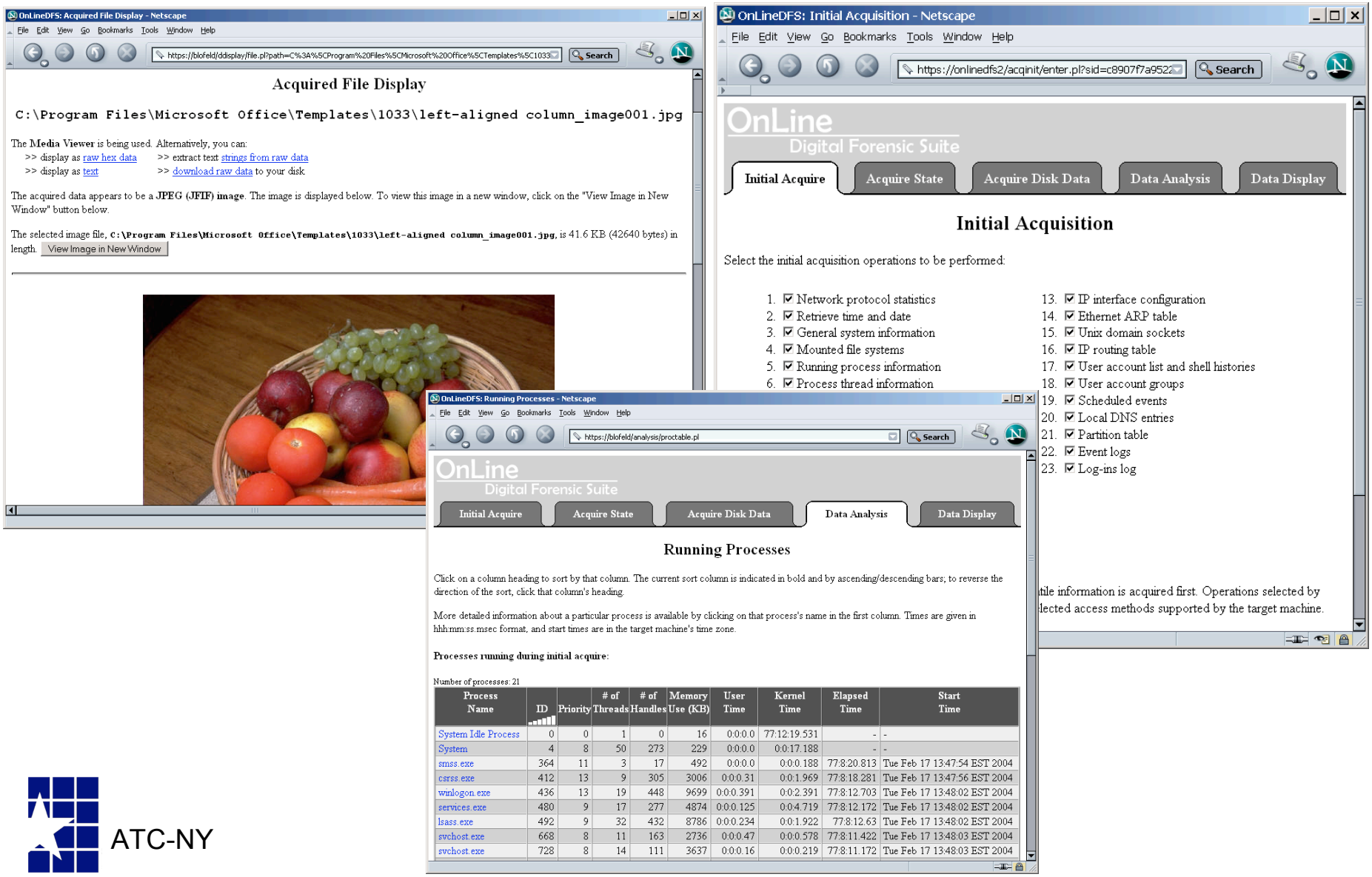
- User specifications

```
user {
    name = next("Namelist");
    class = web, mail;
    tools = pine, firefox | mozilla, spamassassin;
    timeframe = 2005-03-21 : 2005-05-12;
    onlinehistory = daily*4;
  }
```

- Event specifications

```
event {
    label = "syslog-nuke";
    probability = 1.0;
    command = "cd /var/logs; rm syslog";
    time = 5min after event("install-backdoor");
  }
```

FALCON Lab Execution Environment – OnlineDFS



ATC-NY

FALCON Lab Evaluation Tool

- Help evaluate performance, allows instructor to target specific parts of the investigation
- Generate performance statistics and detect some cheating and collusion.
- Looks for “flags” in OnLineDFS’ logs
  - Verify certain directories or processes were examined
  - Ensure certain “out of bounds” data were *not* examined
  - Types of things to examine
    - Pattern matches via regular expressions
    - Sequences of events
    - Time intervals (e.g., How much time did students spend on the lab max, min, and average?)

Summary and Future

- Objective
  - Give students hands-on lab experience
  - Get feedback on from students
  - Improve the process
- Results
  - Identified lab weaknesses
  - Designed framework for automation to reduce effort required for concept and implementation
- FALCON will help with the creation and modification of labs *and* initial evaluation of students' performance and assessing the difficulty level of the lab
- Future: piece it all together!