Automated Digital Evidence Target Definition Using Outlier Analysis and Existing Evidence

By

Brian Carrier and Eugene Spafford

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

Automated Target Definition Using Existing Evidence and Outlier Analysis

Brian D. Carrier

Eugene H. Spafford

CERIAS - Purdue University

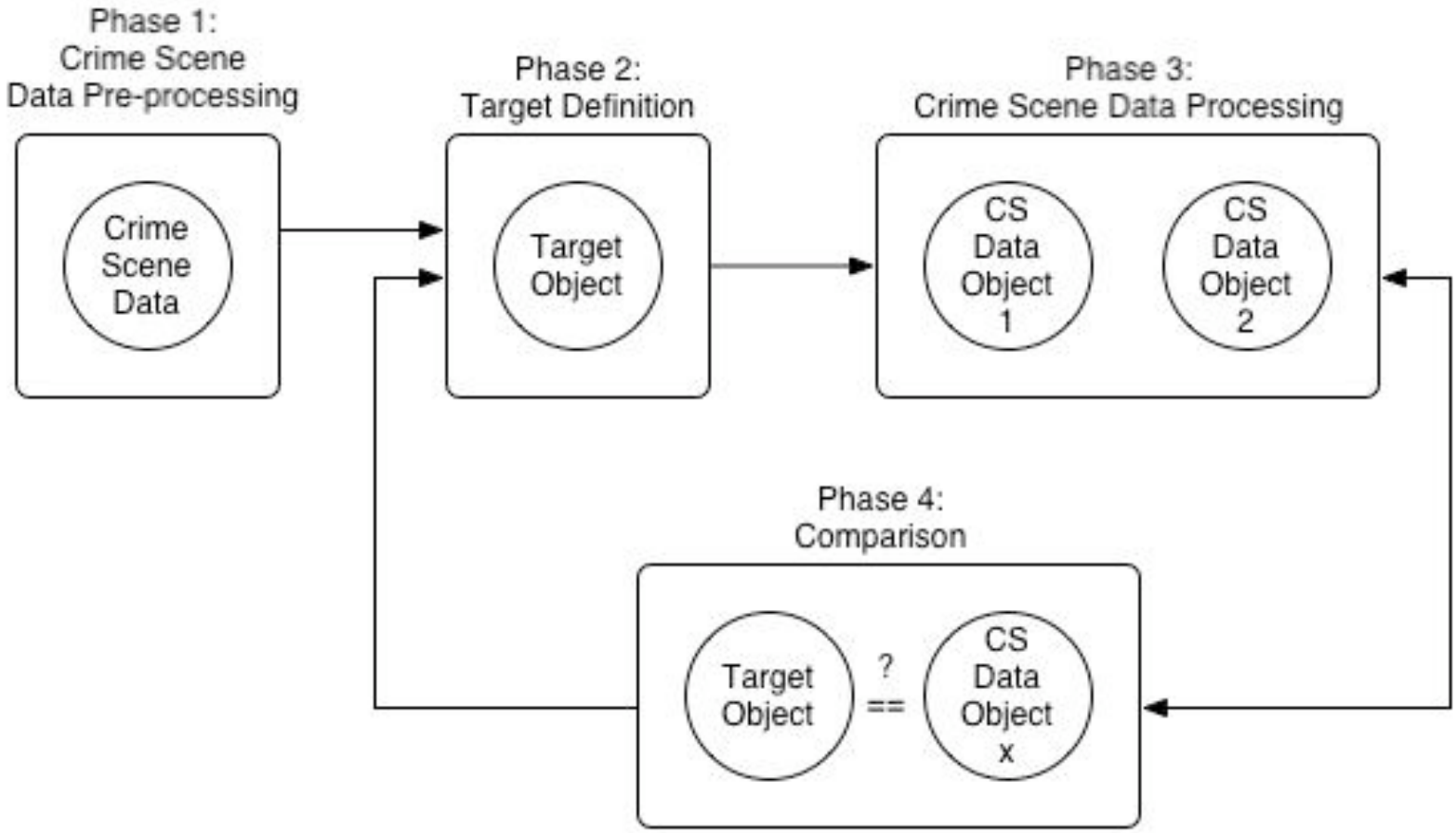
DFRWS 2005

Searching Phases

1. Crime Scene Data Preprocessing

2. Target Definition
   - For what are we looking?
3. Crime Scene Data Processing
   - Process abstraction layers
4. Comparison
   - Are the crime scene data and the target similar?

Searching Phases



How Do We Define Target Objects?

- Develop a hypothesis
  - Experience and training
  - Existing evidence
- Determine what evidence would support or refute hypothesis
- Define the attributes in a target object

Existing Evidence Automation in Autopsy

1. Investigator identifies “evidence”
2. Tool makes suggestions for future searches
3. Tool saves approved searches
4. Investigator selects suggested searches

Suggested Targets

- Files in same parent directory
- Files with same temporal data
- Files with similar names
- Files with same application type
- Files with file name in content
- Files with similar content

Case Study Results

- Honeynet Forensic Challenge
- We find `/dev/ptyp` using experience
- Autopsy suggests similar times, name in content etc.
- Finds `/bin/ps` file and `/usr/man/.Ci` directory

New Topic: File Hiding Techniques

- Change MAC times
- Use similar names
- Pad data to maintain size or CRC
- Make “innocent looking” directory
- Special characters

Make file characteristics “fit in”

Spatial Outlier

- Outliers objects are “grossly different” from all other objects
- Spatial outliers are “grossly different” from local objects
- Hypothesis: Hidden files could be spatial outliers in their parent directory

Variant of Iterative Z Algorithm

Lu, Chen, & Kou - 2003 IEEE CDM

1. Compute average attribute value (g) for each directory
2. Compute distance (h_i) from each file to g
3. Compute mean (μ) and std dev (σ) of h_i
4. Standardize each h_i :

$$y_i = \left| \frac{h_i - \mu}{\sigma} \right|$$

- If largest y_i is > θ (2 or 3), it is an outlier

Single Attribute Results

	/ partition			/usr/ partition		
Attribute	**Hits**	**Accuracy**	**Comp**	**Hits**	**Accuracy**	**Comp**
App Type	1.01%	10.00%	14.00%	9.15%	3.08%	18.58%
Block	6.93%	6.22%	60.00%	3.96%	2.97%	7.74%
C-time	2.24%	26.92%	84.00%	1.27%	54.44%	45.51%
M-time	0.75%	5.77%	6.00%	8.26%	7.63%	41.49%
Size	4.78%	2.41%	16.00%	5.84%	3.22%	12.38%

	Combined		
Attribute	**Hits**	**Accuracy**	**Comp**
App Type	7.14%	3.33%	17.96%
Block	4.69%	4.15%	14.75%
C-time	1.51%	44.37%	50.67%
M-time	6.41%	7.58%	36.73%
Size	5.58%	3.05%	12.87%

/: 6,951 files and 50 (0.72%) involved

/usr/: 21,267 files and 323 (1.49%) involved

Results Summary

- Trojan exec on `/` not found, but was found on `/usr/` because of starting block
- Many false positives were obvious: README files
- Generic Windows System:
  - 2.58% files identified (5.07% on honeypot)
  - 100% false positive rate

Multiple Attributes

1. Standardize all attributes
2. Compute average attribute for each directory (g) and distance for each file (hi)
3. Compute directory mean and variance-covariance

$$\mu_S = \frac{1}{|NN(x_i)|} \sum_{x \in NN(x_i)} h(x)$$

$$\Sigma_s = \frac{1}{|NN(x_i)|} \sum_{x \in NN(x_i)} [h(x) - \mu_s][h(x) - \mu_s]^T$$

Multiple Attributes (2)

4. Compute Mahalonobis distance for each file:

$$a_i = (h(x_i) - \mu_s)^T \Sigma (h(x_i) - \mu_s)$$

5. Standardize distances and determine if largest is > 3

- Lu, Chen, Kou - 15th IEEE Conference on Tools with AI

Multiple Attribute Results

	/ partition			/usr/ partition		
Attribute	**Hits**	**Accuracy**	**Comp**	**Hits**	**Accuracy**	**Comp**
C- and M-time	0.30%	85.71%	36.00%	0.56%	23.33%	8.67%
C-, M-time, and Size	0.22%	53.33%	16.00%	0.32%	43.28%	8.98%
C-, M-time, Size, and Block	0.22%	53.33%	16.00%	0.22%	31.91%	4.64%
C-, M-time, Size, Block and App Type	0.10%	57.14%	8.00%	0.15%	38.71%	3.72%

	Combined		
Attribute	**Hits**	**Accuracy**	**Comp**
C- and M-time	0.50%	32.62%	12.33%
C-, M-time, and Size	0.29%	45.12%	9.92%
C-, M-time, Size, and Block	0.22%	37.10%	6.17%
C-, M-time, Size, Block and App Type	0.13%	42.11%	4.29%

Clean Windows:

C&M: 0.63%

C,M,&S: 0.54%

Hidden Directories

- Calculate average attribute value for files in a directory
- Compare with other directories at same “level”
- Detect outliers using “iterative Z”

Directory Outlier Results

	/ partition			/usr/ partition		
Attribute	**Hits**	**Accuracy**	**Comp**	**Hits**	**Accuracy**	**Comp**
App Type	0.00%	0.00%	n/a%	7.44%	0.00%	0.00%
C-time	2.25%	0.00%	n/a%	1.65%	0.00%	0.00%
M-time	1.12%	0.00%	n/a%	1.38%	0.00%	0.00%
Size	1.12%	0.00%	n/a%	3.77%	0.00%	0.00%

	Combined		
Attribute	**Hits**	**Accuracy**	**Comp**
App Type	6.88%	0.00%	0.00%
C-time	1.70%	0.00%	0.00%
M-time	1.36%	0.00%	0.00%
Size	3.57%	0.00%	0.00%

- `/` had 89 and 0 hidden directories
- `/usr/` had 1,088 and 1 hidden directory
- Was found when $\theta = 2$
- 0.78% on Windows System

Conclusions

- Implemented a tool to suggest and save searches
- Implemented outlier analysis algorithms to find hidden & new files

- What is human error rate?
- Honeypot is not an ideal case study (little user activity before and after incident)

Future Work

- Investigate human error rate
- Identify which techniques work better with different types of incidents
- Incorporate data mining with other analysis techniques - keyword searching, hashes etc.

Brian Carrier

carrier@cerias.purdue.edu

www.cerias.purdue.edu