Reconstructing Alert Trees for Cyber Triage

Eric Ficke
PhD Dissertation Defense
The University of Texas at San Antonio
April 13, 2022

Committee Co-Chairs
Prof. Ravi Sandhu
Prof. Shouhuai Xu

Committee Members
Prof. Xiaoyin Wang
Prof. Greg White
Prof. Mimi Xie
Publications

• Published


• Manuscripts to be submitted for review


Dissertation Outline

• Introduction

• APIN: Alert Path Identification in Computer Networks

• AutoCRAT: Automatic Cumulative Reconstruction of Alert Trees

• Alert Tree Reduction and Visualization

• Conclusion
Introduction

Chapter 1
Background – Alert Trees

- Cyber Triage (Network-level)
  - Alert prioritization
  - Alert correlation
  - Attack lifecycle

- Attack Prediction
  - Attack graphs / trees / paths
  - Vulnerability graphs
Motivation

• Alert volume
  • Unrealistically low in ad hoc datasets
  • Overwhelms human analysis in real data

• Alert graph / tree / path formalization
  • Varies by usage
  • Depends on spatial and temporal dependencies
2. Alert Path Identification (APIN)
   - Alert path reconstruction
   - Threat score (TS) ranking

3. Cumulative Reconstruction (AutoCRAT)
   - Alert tree reconstruction
   - Alternative path reconstruction method
   - Asymptotic and real analysis

4. Reduction and Visualization
   - Mitigates emergent problem of tree size
Dissertation Outline

• Introduction

• APIN: Alert Path Identification in Computer Networks

• AutoCRAT: Automatic Cumulative Reconstruction of Alert Trees

• Alert Tree Reduction and Visualization

• Conclusion
APIN: Alert Path Identification in Computer Networks

Chapter 2
Motivation: Cyber Triage

- Time sensitive
- Resource intensive
- Error prone
- Large search space
Contributions

• Attack Tracking
  • Alert paths show footprints between victim computers
  • Spatiotemporal path reconstruction method

• Heuristics
  • Threat score shows attacker effort
  • (Actual compromise may vary)
APIN Framework

Inputs

NIDS Alerts

ITS

Guided Path Identification

Known Target

Automatic Path Identification

Outputs

CTS

Ranked Attack Paths

NIDS: Network Intrusion Detection System
ITS: Independent Threat Score
CTS: Composite Threat Score
Metric: Independent Threat Score

• Input:
  • $A_{in}$ alert types (inbound)
  • $A_{out}$ alert types (outbound)

• Terms:
  • $D_{in} = |A_{in}|$
  • $D_{out} = |A_{out}|$
  • $S_{in} = |A_{in}| \sqrt{\prod_{a \in A_{in}} |a|}$

$$ITS = \frac{3}{\sqrt[3]{D_{in} \cdot D_{out} \cdot S_{in}}}$$

“$D$” represents alert diversity
“$S$” represents alert scale (by type)
Methods (Alert Path Identification)

Approach: breadth-first search in reverse-chronological order

Not a valid path: Violates temporal attack dependency

Edges indexed by timestamp
Preliminary Analysis

- Scans (high volume, low threat)
  - Prioritize inbound alerts
- Highly connected nodes (high volume, unclear threat)
  - Causes exponential graph growth
- Blacklist nodes
  - Restricts path identification
  - Leaves nodes unmonitored

High-granularity network segmentation improves performance significantly
Metric: Weighted Independent Threat Score

• Input:
  • $A_{in}$ alert types (inbound)
  • $A_{out}$ alert types (outbound)

• Terms:
  • $D_{in} = |A_{in}|$
  • $D_{out} = |A_{out}|$
  • $S_{in} = |A_{in}| \sqrt{\prod_{a \in A_{in}} |a|}$
  • $W = w_1 + w_2 + w_3$

$ITS = \sqrt{W w_1 \cdot D_{in}^{w_2} \cdot S_{in}^{w_3}}$

“$D$” represents alert diversity
“$S$” represents alert scale (by type)
 Metric: Composite Threat Score

\[ \text{CTS} = \sum_{n \in N} \text{ITS}(n) \]
Preliminary Results: DARPA ’99

- Notable paths, using queries from top 5 nodes

<table>
<thead>
<tr>
<th>Origin</th>
<th>Composite Threat Score</th>
<th>Length (#edges)</th>
<th>Notable Alerts</th>
</tr>
</thead>
<tbody>
<tr>
<td>172.16.116.201</td>
<td>3.83</td>
<td>1</td>
<td>Windows 95 Malware</td>
</tr>
<tr>
<td>209.67.29.11</td>
<td>3.43</td>
<td>1</td>
<td>Windows 95 Malware</td>
</tr>
<tr>
<td>172.16.116.194</td>
<td>3.45</td>
<td>1</td>
<td>Windows 95 Malware</td>
</tr>
<tr>
<td>207.25.71.141</td>
<td>3.41</td>
<td>1</td>
<td>Windows 95 Malware</td>
</tr>
<tr>
<td>192.168.1.30</td>
<td>3.41</td>
<td>1</td>
<td>Public SNMP Access</td>
</tr>
<tr>
<td>172.16.112.5</td>
<td>3.41</td>
<td>1</td>
<td>Public SNMP Access</td>
</tr>
<tr>
<td>206.132.25.51</td>
<td>3.36</td>
<td>1</td>
<td>Windows 95 Malware</td>
</tr>
</tbody>
</table>

Hidden IPs are repeated from higher-ranked paths.

Artificial datasets need multi-step attacks.
## Results: CSE-CIC-IDS2018

<table>
<thead>
<tr>
<th>[Path Origin, Path Target]</th>
<th>Composite Threat Score</th>
<th>Length (#Edges)</th>
<th>Notable Alerts</th>
</tr>
</thead>
<tbody>
<tr>
<td>103.47.124.154 54.172.47.69</td>
<td>34.31</td>
<td>4</td>
<td>EternalBlue (WannaCry) NAT Traversal</td>
</tr>
<tr>
<td>172.31.67.54 52.87.201.4</td>
<td>33.60</td>
<td>3</td>
<td>EternalBlue (WannaCry) NAT Traversal</td>
</tr>
<tr>
<td>71.6.165.200 172.31.64.78</td>
<td>26.42</td>
<td>3</td>
<td>Blacklisted IP group SQL Scan</td>
</tr>
<tr>
<td>77.222.106.20 172.31.66.112</td>
<td>21.30</td>
<td>3</td>
<td>EternalBlue (WannaCry) SMB Share Access</td>
</tr>
<tr>
<td>172.31.64.78 172.31.0.2</td>
<td>21.11</td>
<td>1</td>
<td>Suspicious DNS Query</td>
</tr>
</tbody>
</table>
# Results: CSE-CIC-IDS2018

<table>
<thead>
<tr>
<th></th>
<th>IP Addresses</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1.</strong></td>
<td>103.47.124.154 → 172.31.67.46 → 103.68.10.188 → 172.31.66.112 → 54.172.47.69</td>
</tr>
<tr>
<td><strong>2.</strong></td>
<td>172.31.67.54 → 212.174.232.94 → 172.31.64.46 → 52.87.201.4</td>
</tr>
<tr>
<td><strong>3.</strong></td>
<td>71.6.165.200 → 172.31.64.71 → 149.255.35.24 → 172.31.64.78</td>
</tr>
<tr>
<td><strong>4.</strong></td>
<td>77.222.106.20 → 172.31.67.46 → 103.68.10.188 → 172.31.66.112</td>
</tr>
<tr>
<td><strong>5.</strong></td>
<td>172.31.64.78 → 172.31.0.2</td>
</tr>
</tbody>
</table>

**EternalBlue NAT Traversal**

**Blacklisted IP group SQL Scan**

**EternalBlue SMB Share Access**

**Suspicious DNS Query**
Results: CSE-CIC-IDS2018

1. 103.47.124.154 → 172.31.67.46 → 103.68.10.188 → 172.31.66.112 → 54.172.47.69
   EternalBlue NAT Traversal

4. 77.222.106.20
   EternalBlue SMB Share Access

2. 172.31.67.54 → 212.174.232.94 → 172.31.64.46 → 52.87.201.4
   EternalBlue NAT Traversal

3. 71.6.165.200 → 172.31.64.71 → 149.255.35.24 → 172.31.64.78
   Blacklisted IP group SQL Scan

5. 172.31.0.2
   Suspicious DNS Query
Results: CSE-CIC-IDS2018

- IP addresses and connections:
  - 54.172.47.69
  - 103.68.10.188
  - 172.31.64.112
  - 172.31.67.46
  - 149.255.35.24
  - 52.87.201.4
  - 71.6.165.200
  - 172.31.64.78
  - 172.31.64.71
  - 149.255.35.24
  - 71.6.165.200

- Attack vectors:
  - EternalBlue
  - NAT Traversal
  - SQL Scan
  - Suspicious DNS Query
  - SMB Share Access

- Networks:
  - Internal Network

- Origin and Target:
  - Origin
  - Target
Dissertation Outline

• Introduction

• APIN: Alert Path Identification in Computer Networks

• AutoCRAT: Automatic Cumulative Reconstruction of Alert Trees

• Alert Tree Reduction and Visualization

• Conclusion
AutoCRAT: Automatic Cumulative Reconstruction of Alert Trees

Chapter 3
Motivation: Alert Tree Optimization

• Improve reconstruction
• Identify optimization tradeoffs
• Formalize alert trees
Methods (Path Maintenance)

Approach: maintain every path at all times, merging as they join

Paths grow **sequentially**
Paths remain **independent until linked**
Trees form **spontaneously**
Methods (Tree Reconstruction)

Approach: maintain every path at all times, merging as they join

Paths grow **sequentially**
Paths remain **independent until linked**
Trees form **spontaneously**
## Asymptotic Comparison

<table>
<thead>
<tr>
<th>Action</th>
<th>APIN</th>
<th>AutoCRAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert</td>
<td>$O(1)$</td>
<td>$O(</td>
</tr>
<tr>
<td>Retrieve Paths</td>
<td>$O(</td>
<td>V</td>
</tr>
<tr>
<td>Retrieve Trees</td>
<td>$O(</td>
<td>V</td>
</tr>
<tr>
<td>Reinsert</td>
<td>$O(1)$</td>
<td>$O(</td>
</tr>
<tr>
<td>Database Size</td>
<td>$O(</td>
<td>V</td>
</tr>
</tbody>
</table>

*APIN ranks nodes, while AutoCRAT ranks endpoints and paths.*
Results Comparison

<table>
<thead>
<tr>
<th></th>
<th>APIN</th>
<th>AutoCRAT</th>
<th>APIN-Internal</th>
<th>AutoCRAT-Internal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build DB</td>
<td>29m43s</td>
<td>13h42m41s</td>
<td>9s</td>
<td>35s</td>
</tr>
<tr>
<td>Rank Objects*</td>
<td>49s</td>
<td>1h00m29s</td>
<td>0.28s</td>
<td>5s</td>
</tr>
<tr>
<td>Retrieve Top 100 Paths †</td>
<td>52s</td>
<td>32ms</td>
<td>3s†</td>
<td>23ms†</td>
</tr>
<tr>
<td>Retrieve Top 20 Trees †</td>
<td>52s</td>
<td>3m24s†</td>
<td>3s†</td>
<td>1.97s†</td>
</tr>
<tr>
<td>Coverage (Nodes)</td>
<td>99.6%</td>
<td>100%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coverage (Events)</td>
<td>3.4%</td>
<td>100%</td>
<td>0.6%</td>
<td>0.6%</td>
</tr>
<tr>
<td>Database Size</td>
<td>637 MB</td>
<td>1.1 GB</td>
<td>2.9 MB</td>
<td>2.4 MB</td>
</tr>
</tbody>
</table>

*APIN ranks nodes, while AutoCRAT ranks endpoints and paths.
†These ranks are inferred from their ends (for paths) or root (for trees)
Dissertation Outline

• Introduction

• APIN: Alert Path Identification in Computer Networks

• AutoCRAT: Automatic Cumulative Reconstruction of Alert Trees

• Alert Tree Reduction and Visualization

• Conclusion
Alert Tree
Reduction and Visualization

Chapter 4
Motivation

Facilitate cyber triage by selectively pruning alert trees

• Reduce visual strain
  • “Which nodes can be removed to facilitate tree interpretation?”

• Preserve salient information
  • “What nodes must be kept based on relevant metrics?”
Motivating Example

• This tree (from real data) has 3090 nodes.
• Graphviz is forced to render it at 6% of its original resolution.*

*If you have difficulty reading any of the node labels, that’s exactly the problem we need to solve
Motivating Example

• After reduction, 3090 nodes becomes 40 nodes (98.7% reduction)

So how do we do it???
Alert Tree Reduction Architecture

Input Alert Trees

Annotate Tree

- Merge Similar Sibling Branches
- Truncate Internal Hypotrees

Merge Sibling Leaves

- Reduction 1
- Reduction 2
- Reduction 3
- Reduction 4
- Reduction 5

Output Reduced Trees
Duplicate labels may exist in a tree but not in a path
Terminology (Graph Theory vs Data Structures)
Terminology (Graph Theory vs Data Structures)
Terminology (Graph Theory vs Data Structures)

In Graph Theory:
- $T_1$ is a subtree of $T_2$
- $F$ is a subtree of $T_1$ (or $T_2$)

In Data Structures:
- $F$ is a subtree of $B$ (in either tree)

We need a new term for the relationship $T_1:T_2$ that eliminates ambiguity
• Designate: $T_1$ is a hypotree of $T_2$ ($T_1 \preceq T_2$)
• Designate: $T_2$ is a hypertree of $T_1$ ($T_2 \succeq T_1$)
• Every tree is both a hypotree and a hypertree of itself
• We also designate proper hypotree ($\prec$) and proper hypertree ($\succ$)
Definition: A tree $T_{hypo}$ is a hypotree of a tree $T_{hyper}$ if:

$$\forall n \in T_{hypo}, \exists n' \in T_{hyper}: \forall i \in \{0, 1, ..., |n.ancestors|\}, n.ancestors_i = n'.ancestors_i$$

*Hypertree is derived from hypotree. Refer to the paper for exact detail.*
Merging Sibling Leaves [MSL(A)]

Node labels represent IP addresses.
Duplicate labels may exist in a tree but not in a path within that tree.
Here, node colors show labels (rather than threat score) for ease of understanding.
Merging Similar Sibling Branches
[MSB(A)]

Similar Branches: A set of branches for which all subtrees excluding the branch root exist in both branches.
Merging Sibling Branches & Leaves
[MSL(MSB(A))]
Truncating Hypotrees
[TH(A)]
Truncating Hypotrees & Merging Sibling Leaves

[MSL(TH(A))]
Method Restrictions

• MSL makes some trees similar (because “M2” = “M2”)
  • MSB(MSL(T)) is unsafe (but MSL(MSB(T)) is safe)
  • TH(MSL(T)) is unsafe (but MSL(TH(T)) is safe)

• MSB and TH may target the same branches
  • MSB(TH(T)) ≠ TH(MSB(T))

• The 5 valid reduction schedules:
  1. MSB(T)
  2. MSL(MSB(T))
  3. MSL(T)
  4. MSL(TH(T))
  5. TH(T)

MSL: Merge Sibling Leaves
MSB: Merge Sibling Branches
TH: Truncate Hypotrees
Method Comparisons (Toy Example)

**MSL(A):**
- A
- M2
- B
- C
- D
- E
- F
- G
- H

35% reduction

**MSB(A):**
- A
- M2
- C
- M2
- E
- F
- G
- H

18% reduction

**TH(A):**
- A
- M2
- B
- C
- D
- E
- F
- G
- H

6% reduction

ML: Merge Sibling Leaves
MSB: Merge Sibling Branches
TH: Truncate Hypotrees
Method Comparisons (Toy Example)

**MSL(A):**
- **A**
- **B**
- **C**
- **D**
- **E**
- **M3**

- **35% reduction**

**MSL(MSB(A)):**
- **A**
- **C**
- **M2**
- **E**
- **M3**

- **47% reduction**

**MSL(TM(A)):**
- **A**
- **B**
- **C**
- **D**
- **E**
- **M3**

- **35% reduction**

**MSL: Merge Sibling Leaves**

**MSB: Merge Sibling Branches**

**TH: Truncate Hypotrees**
Visualization

- Black (low threat) -> red (high threat)
  - Min-max normalized
- Merged nodes
  - Color shows highest threat of those merged
Results (Visual):
Forward Tree 204.237.142.47

• Full Tree

• Tree with branches merged (R1)

• Tree with branches and leaves merged (R2)
Metrics

- Visual Strain Reduction (VSR)
- Node Retention (NR)
- Threat Score Retention (TSR)
- Reduction Index (RI)
  - \[ RI = \frac{3}{VSR^{-1} + NR^{-1} + TSR^{-1}} \]
## Results (Numerical)

<table>
<thead>
<tr>
<th>Reduction</th>
<th>Tree Set</th>
<th>VSR</th>
<th>NR</th>
<th>TSR</th>
<th>RI</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MSB</strong></td>
<td>Top 5</td>
<td>0.243</td>
<td>0.539</td>
<td>0.278</td>
<td>0.313</td>
</tr>
<tr>
<td></td>
<td>Random 5</td>
<td>0.352</td>
<td>0.553</td>
<td>0.254</td>
<td>0.349</td>
</tr>
<tr>
<td></td>
<td>Bottom 5</td>
<td>0.433</td>
<td>0.493</td>
<td>0.36</td>
<td>0.42</td>
</tr>
<tr>
<td><strong>MSL</strong></td>
<td>Top 5</td>
<td>0.363</td>
<td>0.577</td>
<td>0.611</td>
<td>0.489</td>
</tr>
<tr>
<td></td>
<td>Random 5</td>
<td>0.282</td>
<td>0.824</td>
<td>0.799</td>
<td>0.499</td>
</tr>
<tr>
<td></td>
<td>Bottom 5</td>
<td>0.791</td>
<td>0.744</td>
<td>0.73</td>
<td>0.754</td>
</tr>
<tr>
<td><strong>TH</strong></td>
<td>Top 5</td>
<td>0.009</td>
<td>1</td>
<td>0.999</td>
<td>0.026</td>
</tr>
<tr>
<td></td>
<td>Random 5</td>
<td>0.037</td>
<td>1</td>
<td>0.983</td>
<td>0.103</td>
</tr>
</tbody>
</table>

MSB: Merge Sibling Branches  
MSL: Merge Sibling Leaves  
TH: Truncate Hypotrees  
VSR: Visual Strain Reduction  
NR: Node Retention  
TSR: Threat Score Retention  
RI: Reduction Index
Dissertation Outline

• Introduction

• APIN: Alert Path Identification in Computer Networks

• AutoCRAT: Automatic Cumulative Reconstruction of Alert Trees

• Alert Tree Reduction and Visualization

• Conclusion
Conclusion

Chapter 5
Discussion

• APIN
  • Relies on network segmentation
  • Dominates maintenance time

• AutoCRAT
  • Relies on ordering assumption
  • Dominates retrieval time

• Reduction improves visualization

eric.ficke@utsa.edu

Images courtesy of pixabay.com and publicdomainvectors.org