Coordinated Scan Detection

Carrie Gates

CA Labs

carrie.gates@ca.com
A Few Definitions to Start….

1. A **target** is a single port at a single IP address.
2. A **scan** is a set of connection attempts from a single source to a set of targets during time interval.
3. A **source** is a computer system from which a scan originates.
4. A **coordinated scan** is a collection of scans from multiple sources where there is a single instigator behind the set of sources.
What is a co-ordinated port scan?
A detector can be designed to detect coordinated TCP port scans against a target network where the scan footprint is either horizontal or strobe with a high detection rate (>= 98%) and a low false positive rate (< 1%) on /16 networks.
Related Work

1. Defining a coordinated scan as having very specific characteristics so that scans can be easily clustered

2. Clustering packets or alerts based on feature similarities using a machine learning approach

3. Manual analysis of network traffic, often aided by visualization approaches, to detect patterns that are representative of coordinated scanning activity
Methodology

1. Develop a model of adversary types
2. Develop a detector based on the model
3. Evaluate the detector
   1. Identify key variables
   2. Model using regression equations
Adversary Model
Adversary Model

- Developed based on:
  - Adversary targets
  - Footprint scan of these targets generates
    \( \forall = < |A|, |P|, \zeta(C), \mathcal{H}(C), \zeta, \kappa > \)
  - 21 adversary footprint patterns identified

We have developed a detector that can detect 9 of the 21 adversary types, where either \( \zeta \) or \( \kappa \) contains at least one subnet.
• Inspired by the set covering problem - find the minimum number of sets that covers the entire space

• Our modification: find the set of scans that maximizes coverage, $\zeta(C)$, while minimizing overlap, $\theta$
Detector

- Coordinated scan recognized in set if:
  1. Set consists of more than one scan, $|S| > 1$
  2. Overlap is acceptably small, $\Theta < Y\%$
  3. Coverage is acceptably large, $\zeta(C) > X\%$
  4. Hit rate is acceptable large, $\mathcal{H}(C) > Z\%$
Algorithm (Altgreedy Portion)

\[ S \leftarrow \text{smallestScan}(A) \]

repeat

\[ i \leftarrow \text{smallestOverlap}(A - \text{rejected}, S) \]

if newlyCoveredIPs( \( S, i \) ) > 0 then

add scan to solution set

else

possibly reject scan

if overlap(\( S \)) > MAXOVERLAP then

\[ i \leftarrow \text{greatestOverlap}(S) \]

\[ S \leftarrow S - \{i\} \]

possibly reject scan

until \( S \cup \text{rejected} == A \)
Algorithm (Detection Portion)

while overlap($S$) > MAXOVERLAP
    $i \leftarrow$ greatestOverlap($S$)
    $S \leftarrow S - i$
end while

while (! isDPS($S$)) && (coverage($S$) > MINCOVERAGE)) do
    gap $\leftarrow$ largest set of contiguous IP addresses not covered in $S$
    $S \leftarrow$ scans in largest subset of $S$ when split into two sets
end while

if isDPS($S$) then
    results $\leftarrow S$
end if
Testing the Algorithm

- Ideal case is real, labeled data
  - Hard to obtain
  - How do you confirm that labels are correct?
  - Red-teaming

- Emulation
  - Uses real data as background noise
  - Uses / restricted to actual scan tools
  - Isolated environment means no legal issues

- Simulation
  - Need to prove that simulation contains no bias
  - Potentially allows greater exploration of space
Experimental Design

- Scans were performed on DETER testbed

- Noise was obtained from four /16 live networks
Identification of Key Variables

- What are the inputs?
  1. Minimum network coverage
  2. Maximum overlap
  3. Number of (noise) scans

- What are the scan characteristics?
  4. Scanning algorithm
  5. Number of scanning sources
  6. Number of ports scanned
<table>
<thead>
<tr>
<th></th>
<th>Values for Key Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Network Coverage</td>
</tr>
<tr>
<td></td>
<td>Network Coverage</td>
</tr>
<tr>
<td></td>
<td>0x10 100</td>
</tr>
<tr>
<td>2</td>
<td>Overlap</td>
</tr>
<tr>
<td></td>
<td>Overlap</td>
</tr>
<tr>
<td></td>
<td>0 10x20</td>
</tr>
<tr>
<td>3</td>
<td>Number of Noise Scans</td>
</tr>
<tr>
<td></td>
<td>Number of Noise Scans</td>
</tr>
<tr>
<td></td>
<td>0 10x100 X 1000</td>
</tr>
<tr>
<td>4</td>
<td>Scanning Algorithm</td>
</tr>
<tr>
<td></td>
<td>Scanning Algorithm</td>
</tr>
<tr>
<td></td>
<td>DScan, NSAT</td>
</tr>
<tr>
<td>5</td>
<td>Number of Scanning Sources</td>
</tr>
<tr>
<td></td>
<td>Number of Scanning Sources</td>
</tr>
<tr>
<td></td>
<td>2 700000 X 100</td>
</tr>
<tr>
<td>6</td>
<td>Number of Scanned Ports</td>
</tr>
<tr>
<td></td>
<td>Number of Scanned Ports</td>
</tr>
<tr>
<td></td>
<td>1 65536 X 5</td>
</tr>
</tbody>
</table>
## Training and Testing Data

| Cov % | Ov % | Algo | Scan Win | | S| | P| | DR | FP |
|------|-----|------|----------|-----|---|---|---|---|---|
| 86   | 0   | 0    | 800      | 39  | 1 | 1.00 | 0.003 |
| 77   | 11  | 1    | 900      | 36  | 5 | 1.00 | 0.006 |
| 64   | 3   | 1    | 200      | 48  | 2 | 1.00 | 0.000 |
| 18   | 17  | 1    | 500      | 64  | 1 | 0.00 | 0.000 |
Regression Model (Detection)

\[ P(\text{co-ordinated scan is detected}) = \frac{e^y}{1 + e^y} \]
\[ y = -1.592 + 0.031 x_1 - 0.003 x_4 + 0.021 x_5 + 0.576 x_6 \]
Regression Model (False Positives)

\[ fp = -0.007494 + 0.00005559 \, x_1 + 0.0004216 \, x_2 \\
+ 0.00005877 \, x_5 + 0.001903 \, x_6 \]

\( x_1 = \text{network coverage} \)
\( x_2 = \text{overlap} \)
\( x_5 = \text{number of sources} \)
\( x_6 = \text{number of ports} \)
**Conclusion: Accept Hypothesis**

<table>
<thead>
<tr>
<th>% Cov</th>
<th>% Ov</th>
<th>Noise</th>
<th></th>
<th>S</th>
<th></th>
<th>P</th>
<th>DR</th>
<th>FP</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0</td>
<td>100</td>
<td>100</td>
<td>5</td>
<td>0.998</td>
<td>0.013</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>100</td>
<td>100</td>
<td>5</td>
<td>0.967</td>
<td>0.008</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>1000</td>
<td>100</td>
<td>5</td>
<td>0.979</td>
<td>0.013</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>100</td>
<td>2</td>
<td>5</td>
<td>0.985</td>
<td>0.008</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>0</td>
<td>100</td>
<td>100</td>
<td>1</td>
<td>0.980</td>
<td>0.006</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>20</td>
<td>100</td>
<td>100</td>
<td>5</td>
<td>0.967</td>
<td>0.017</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>20</td>
<td>100</td>
<td>100</td>
<td>5</td>
<td>0.998</td>
<td>0.022</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>20</td>
<td>1000</td>
<td>100</td>
<td>5</td>
<td>0.979</td>
<td>0.022</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>20</td>
<td>100</td>
<td>2</td>
<td>5</td>
<td>0.985</td>
<td>0.016</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>20</td>
<td>100</td>
<td>100</td>
<td>1</td>
<td>0.980</td>
<td>0.014</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
How to Game My Detector

1. Do not scan a contiguous space
   - E.g., all existing hosts might not be contiguous
   - But… can “compress” non-existing hosts to generate contiguous space - *might* address this issue

2. Scan less than 95% of contiguous space
   - Hit rate for algorithm is set at >= 95%
   - Need further work to determine lower bound

3. Distribute scans from each source over enough time

4. Make sure sources are not detected by single-source scan detection algorithm
What is the Effect of Time?

- Time is the wrong variable
- How well does this work when deployed?
  - How much of each scan is required before recognizing a coordinated scan?
  - How many scans are required before the coordinated scan is detected?
  - How should the sliding window be implemented?
Key Contributions

1. Adversary model
   - Provides an enumeration of the possible adversary types in this space

2. Detection algorithm
   - High detection rate and low false positive rate under certain (known) circumstances