# Coordinated Scan Detection

Carrie Gates

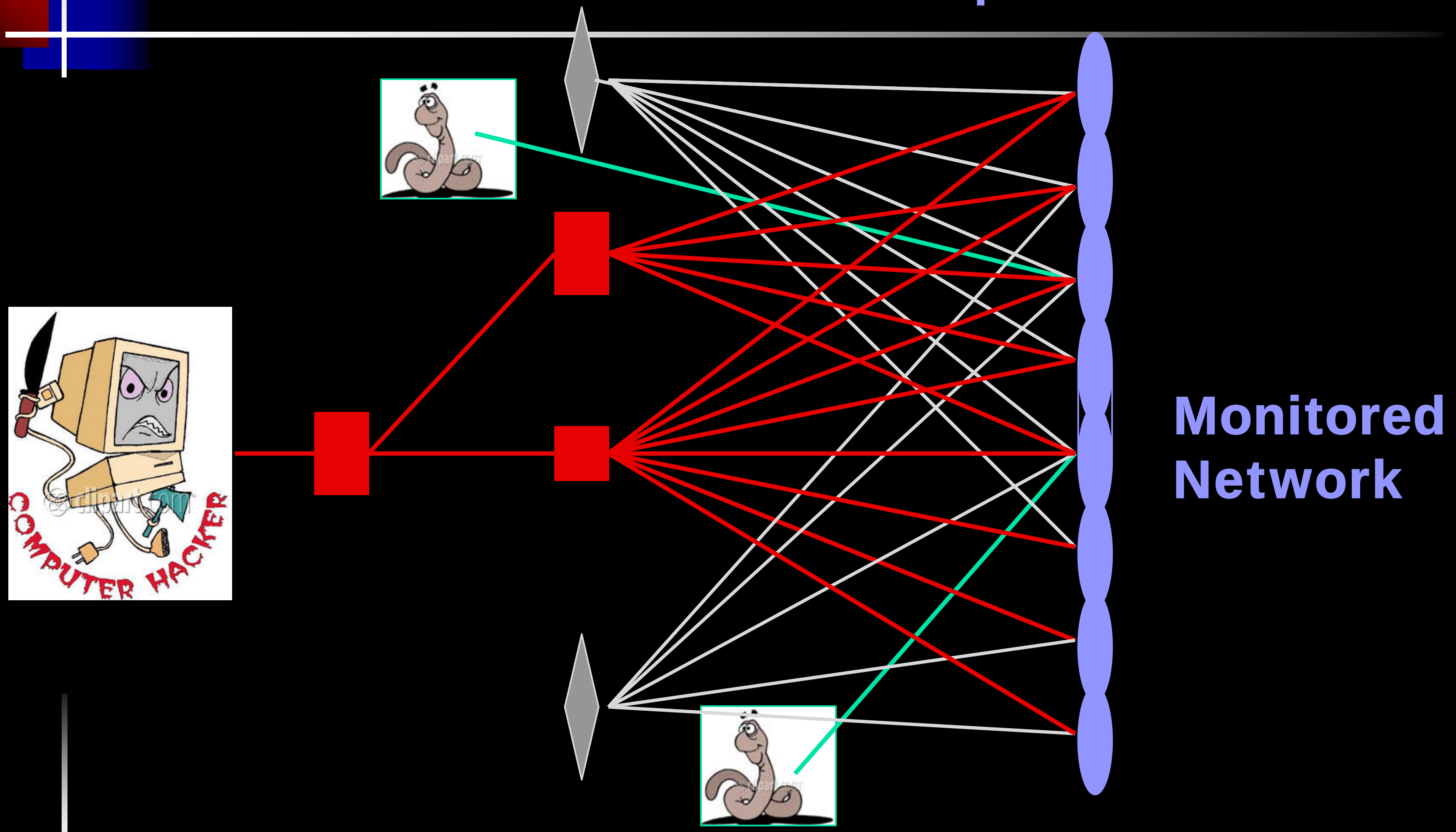
CA Labs

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?



# Hypothesis

*A detector can be designed to detect co-ordinated 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.*

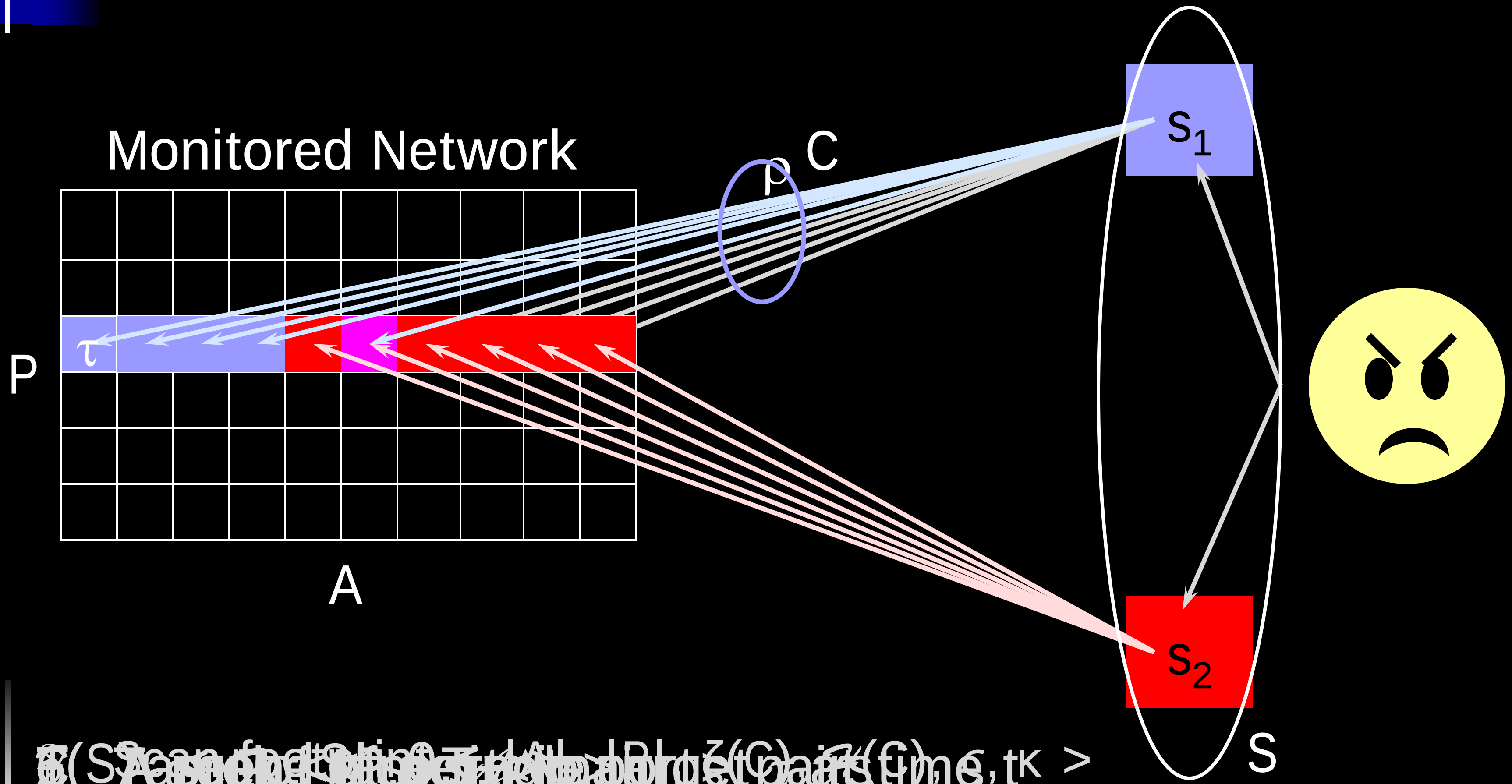
# Related Work

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

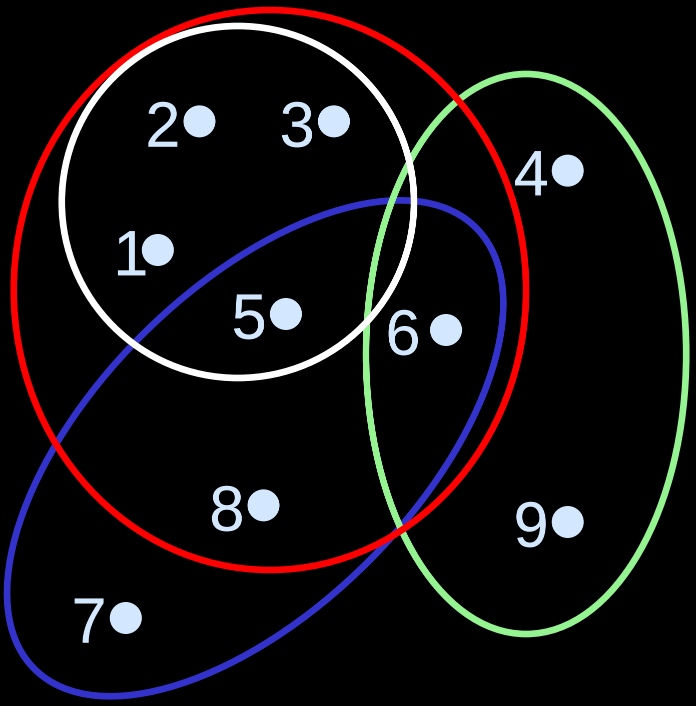


# Adversary Model

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

We have developed a detector that can detect 9 of the 21 adversary types, where either ς or κ contains at least one subnet.

# Detector




- 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, ζ(C), while minimizing overlap, θ

# 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* ← smallestScan(*A*)
repeat
    *i* ← smallestOverlap(*A* – *rejected*, *S*)
    if newlyCoveredIPs( *S*, *i* ) > 0 then
      add scan to solution set
    else
      possibly reject scan
    if overlap(*S*) > MAXOVERLAP then
      *i* ← greatestOverlap(*S*)
      *S* ← *S* – {*i*}
      possibly reject scan
until *S* U *rejected* == *A*

# Algorithm (Detection Portion)

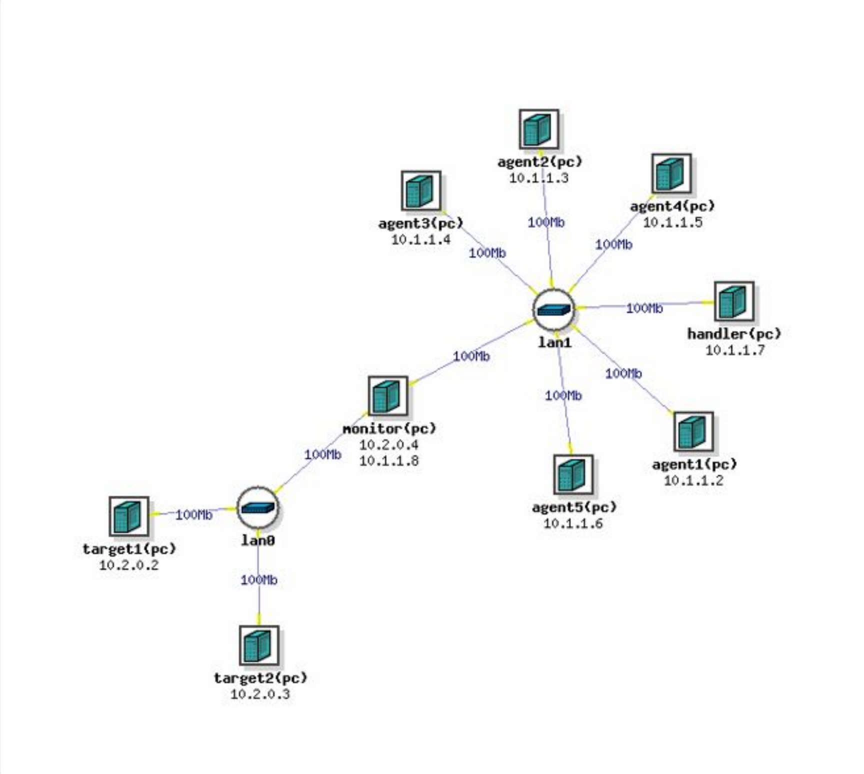
```
while overlap(S) > MAXOVERLAP
        i ← greatestOverlap(S)
        S ← S – i
end while
while (! isDPS(S)) && (coverage(S) > MINCOVERAGE)) do
        gap ← largest set of contiguous IP addresses not covered in S
        S ← scans in largest subset of S when split into two sets
end while
if isDPS(S) then
        results ← 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

# Values for Key Variables

| | | | |
|---|---|---|---|
| 1 | Network Coverage | ~~0~~ *10* | 100 |
| 2 | Overlap | 0 | ~~100~~ *20* |
| 3 | Number of Noise Scans | ~~0~~ *100* | ~~∞~~ *1000* |
| 4 | Scanning Algorithm | DScan, NSAT | |
| 5 | Number of Scanning Sources | 2 | ~~700000~~ *100* |
| 6 | Number of Scanned Ports | 1 | ~~65536~~ *5* |

# Training and Testing Data

| Cov % | Ov % | Algo<br>0 - NSAT<br>1 - DScan | 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 |

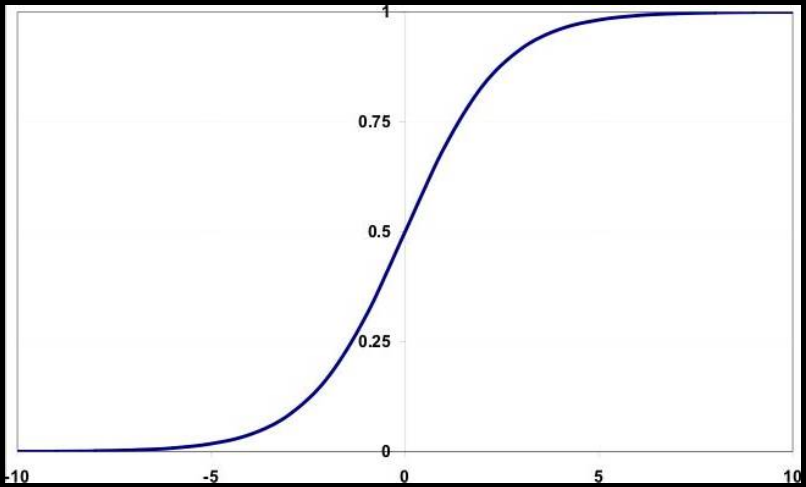
# Regression Model (Detection)

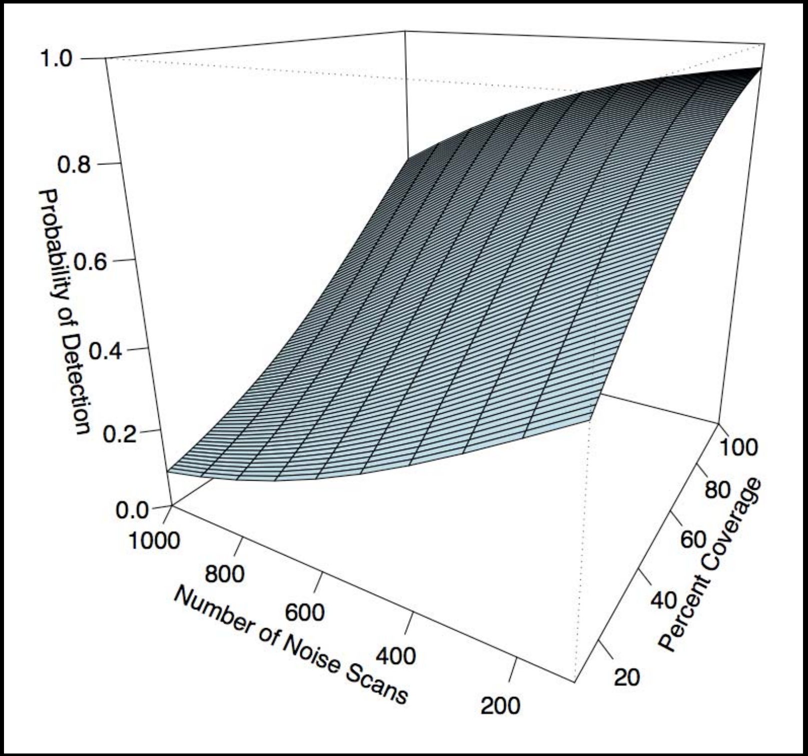
$P$(co-ordinated scan is detected) = $e^y / (1 + e^y)$

$y = -1.592 + 0.031\ x_1$

$-\ 0.003\ x_4 + 0.021\ x_5$

$+\ 0.576\ x_6$



# Regression Model (False Positives)

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

$x_1$ = network coverage

$x_2$ = overlap

$x_5$ = number of sources

$x_6$ = number of ports

# Conclusion: Accept Hypothesis

| % Cov | % Ov | Noise | \|S\| | \|P\| | DR | FP |
|---:|---:|---:|---:|---:|---:|---:|
| 100 | 0 | 100 | 100 | 5 | 0.998 | 0.013 |
| 10 | 0 | 100 | 100 | 5 | 0.967 | 0.008 |
| 100 | 0 | 1000 | 100 | 5 | 0.979 | 0.013 |
| 100 | 0 | 100 | 2 | 5 | 0.985 | 0.008 |
| 100 | 0 | 100 | 100 | 1 | 0.980 | 0.006 |
| 10 | 20 | 100 | 100 | 5 | 0.967 | 0.017 |
| 100 | 20 | 100 | 100 | 5 | 0.998 | 0.022 |
| 100 | 20 | 1000 | 100 | 5 | 0.979 | 0.022 |
| 100 | 20 | 100 | 2 | 5 | 0.985 | 0.016 |
| 100 | 20 | 100 | 100 | 1 | 0.980 | 0.014 |

# 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