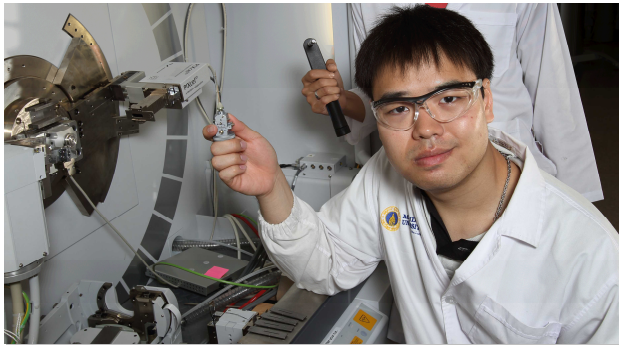
SAND2016-12961C





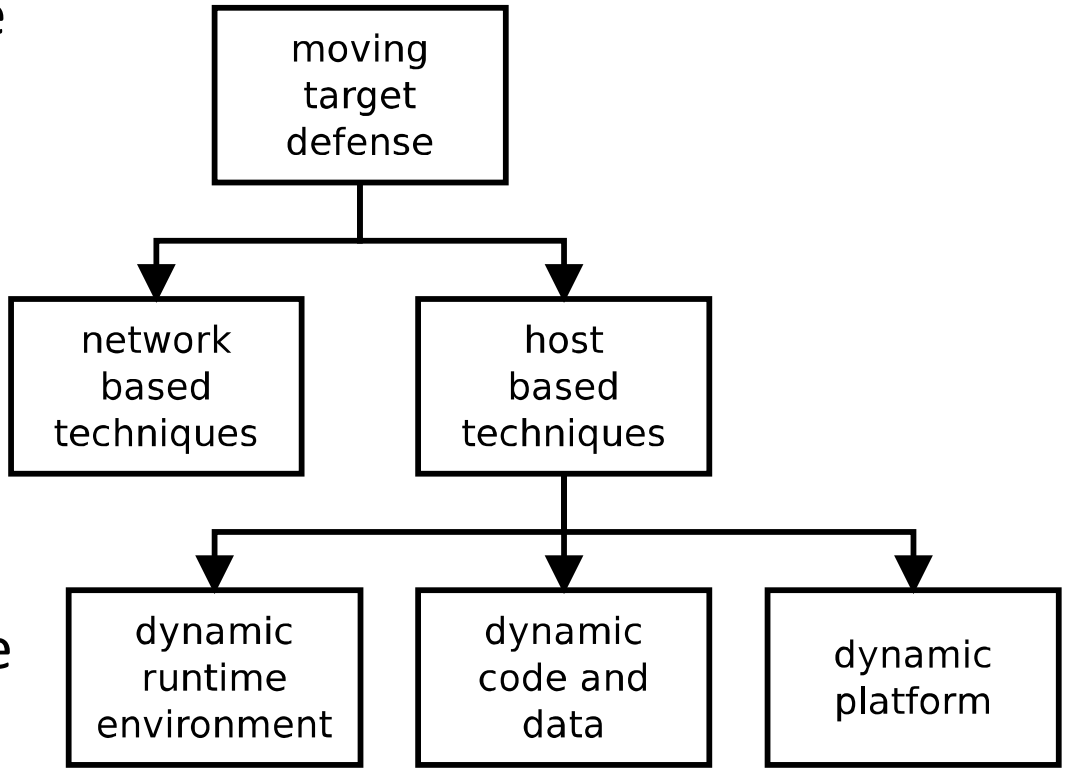
A Zoning Algorithm for Dynamic Cyber Zone Defense

Marci McBride [mmcbri@sandia.gov, (505)609-4303]
Rob Mitchell [rrmitch@sandia.gov]

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXXXP

Introduction

- Concept of static cyber zone defense was introduced by Robert Mitchell.
- Dynamic zoning connects the concept of cyber zone defense to the realm of moving target defense.
- Dynamic cyber zone defense is a network-based moving target defense



Hamed Okhravi's Moving target defense taxonomy

Formulation

Sandia National Laboratories

- Previous work provides
  - Probability of Compromise
  - Probability of Reachback
  - Math model and simulation to predict its effectiveness.
- In order to limit malware communication without interfering with useful work hosts can request additional network visibility on demand.
- Our proposed algorithm will broker these requests and grant the network visibility required to accomplish the mission while thwarting or disrupting cyber attackers.

Metrics of Interest

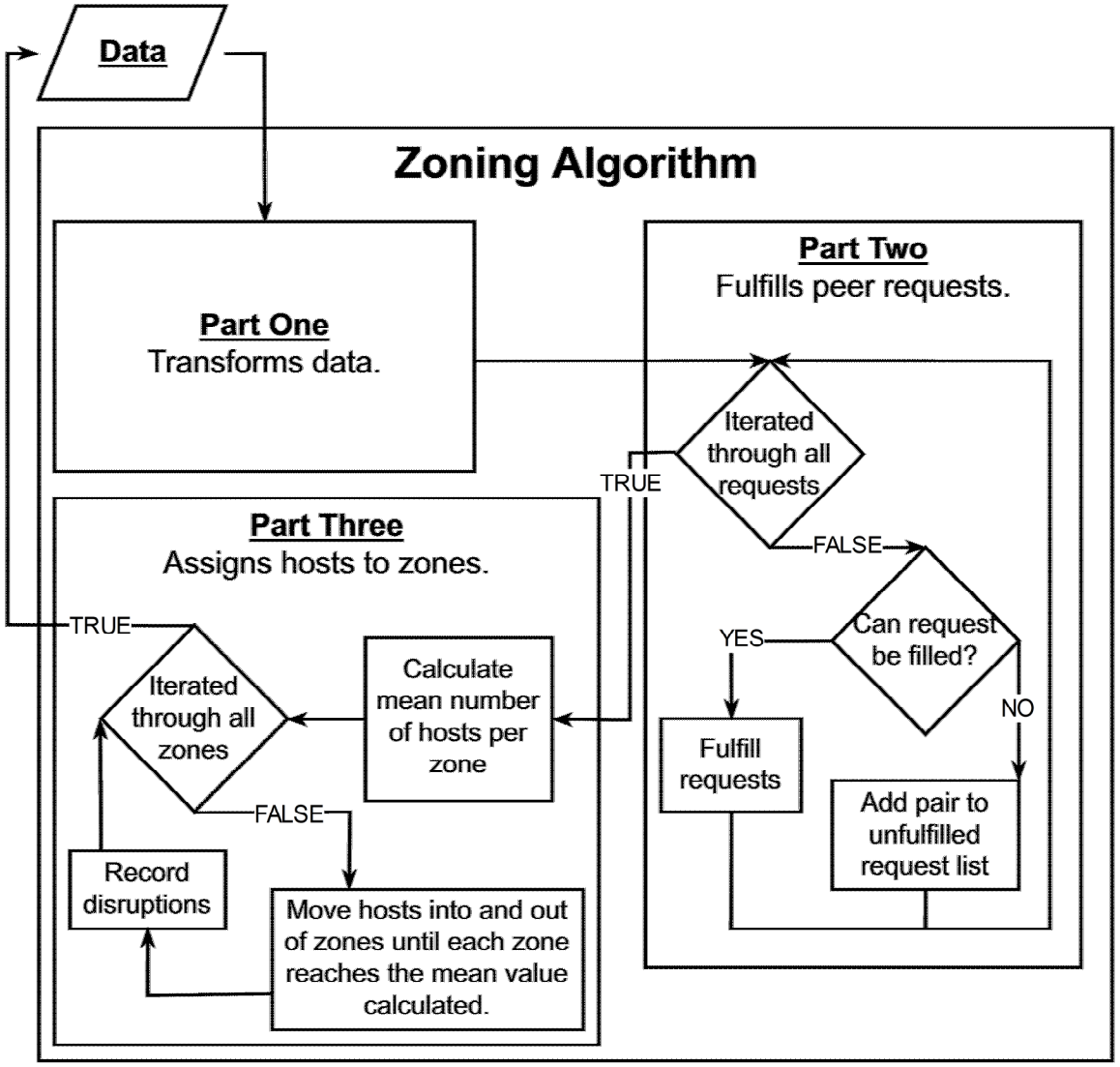
Sandia National Laboratories

The metrics used to gauge the quality of our solution are the following.

- Average number of hosts assigned to a zone
- Average number of zones a host is assigned to
- Number of disruptions

The Algorithm

We utilized modular programing and divided the problem into three separate parts. This approach increased reliability, readability, and maintainability.



Fulfilling Peer Requests

Sandia National Laboratories

Requires five inputs:

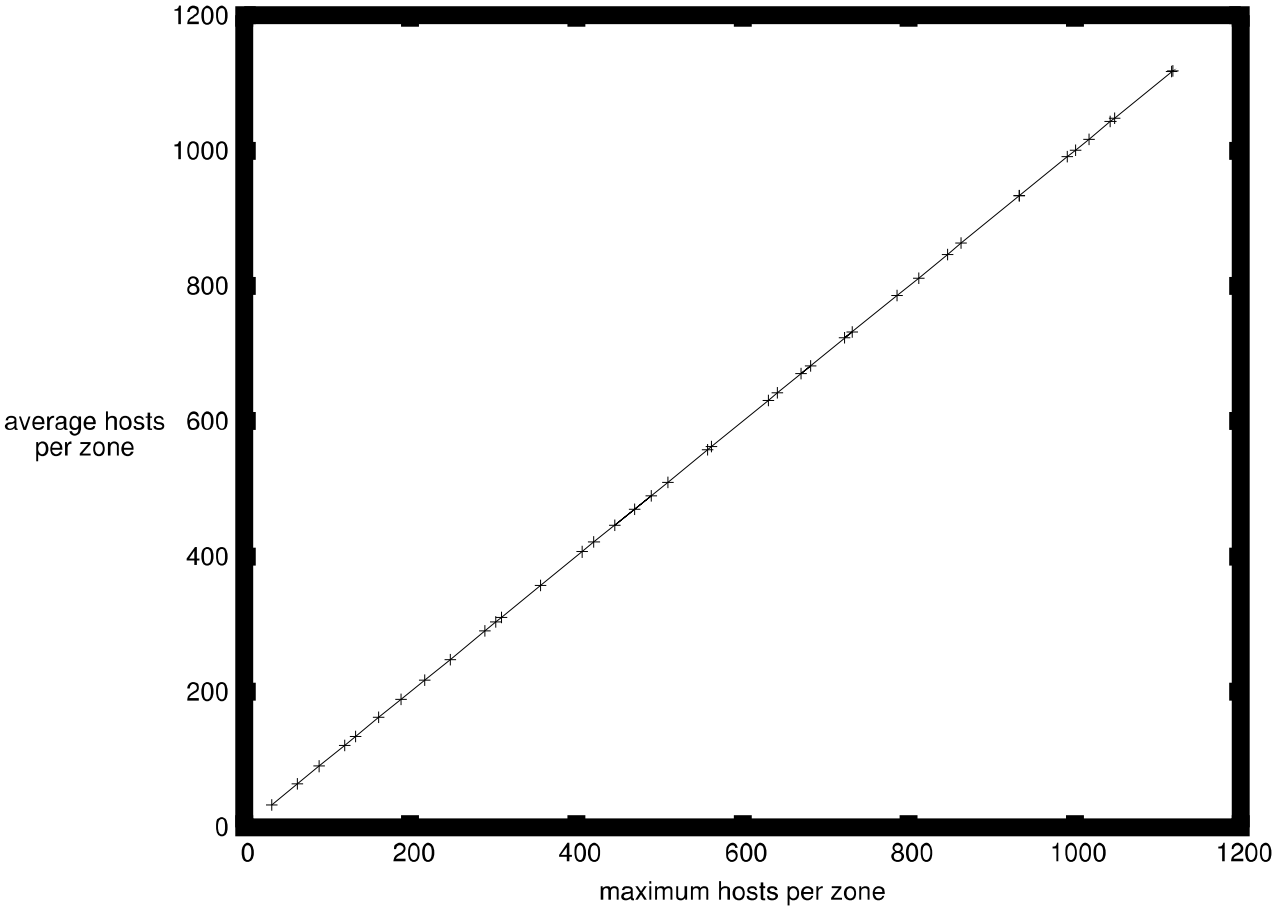
- reqs is the list of peer requests
- mhpz is the maximum number of hosts per zone
- mzph is the maximum number of zones per host
- mz is the maximum number of zones
- ad is the assignment dictionary

Algorithm 1 Peer Request Fulfillment

```
1:  function FULFILL_REQUESTS(reqs, mhpz, mzph, mz, ad)
2:      disruptions ← 0
3:      flag ← 0
4:      for p in reqs do
5:          if p_0 == p_1 then
6:              continue
7:          end if
8:          for hl in ad.values() do
9:              if hl.count(p_0) and hl.count(p_1) then
10:                 flag ← 1
11:                 break
12:             end if
13:         end for
14:         if flag then
15:             flag ← 0
16:             continue
17:         end if
18:         if |ad.keys()| < mz then
19:             ad.update(new_zone_id(ad): [p_0, p_1])
20:             continue
21:         end if
22:         smallest_zone_size ← MAX_ZONE_SIZE
23:         for zone in ad do
24:             if |ad[zone]| < smallest_zone_size then
25:                 smallest_zone ← zone
26:                 smallest_zone_size ← |ad[zone]|
27:             end if
28:         end for
29:         for host in p do
30:             if host not in ad[smallest_zone] then
31:                 ad[smallest_zone].append(host)
32:             end if
33:             if |ad[smallest_zone]| > mhpz then
34:                 ad[smallest_zone].pop(0)
35:                 disruptions += 1
36:             end if
37:             remove_if_necessary(host, ad, mzph)
38:         end for
39:     end for
40:     return disruptions
41: end function
```

Algorithm's Performance

Steady State Metrics



Average hosts per zone versus maximum hosts per zone.

Algorithm's Performance

We found that the average zones per host approached a function of three system parameters which we call α.

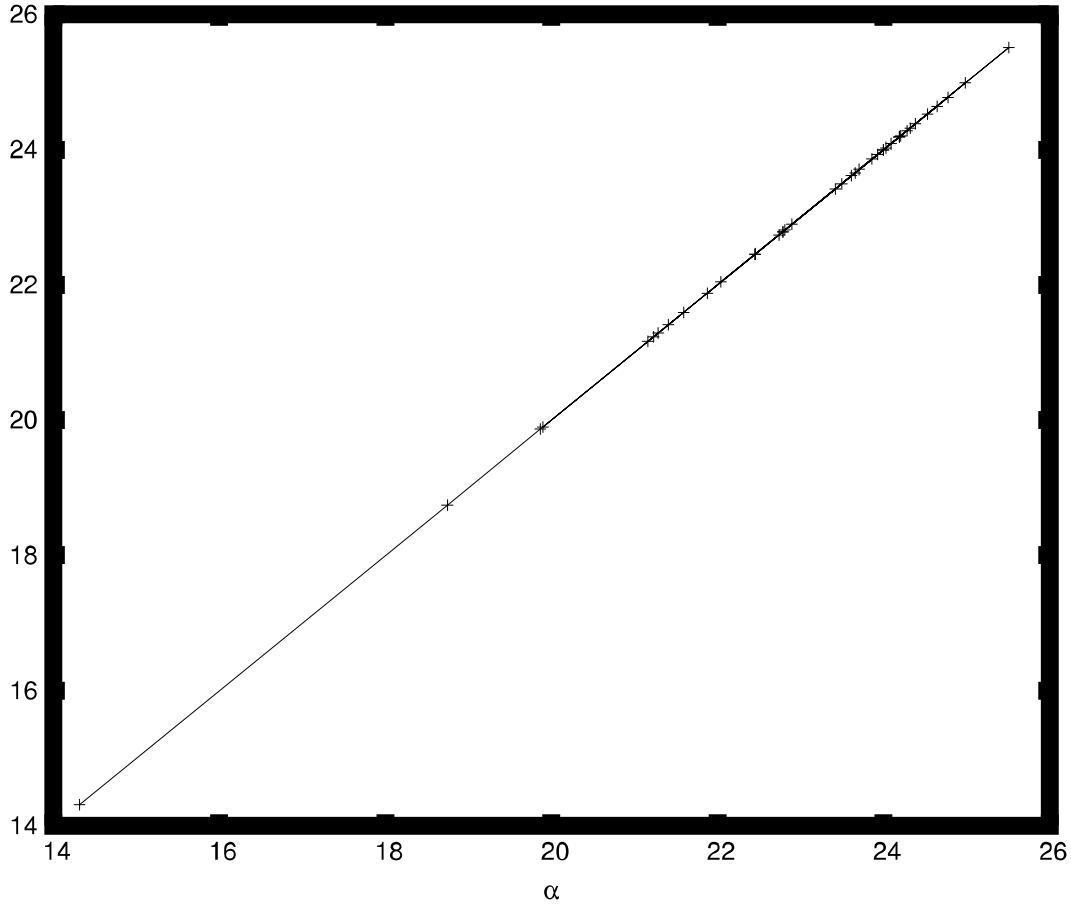
$$\alpha = \frac{(b_{max} \cdot a_{max})}{h}$$

b_{max} = Maximum zones

a_{max} = Maximum hosts per zone

h = Number of hosts

Steady State Metrics



Average zones per host versus α

Algorithm's Performance

We propose the following theoretical model for the average number of disruptions per day for a converged system:

$\bar{d}$ = u · P (PRPH per day)

= u · CDF (λ, 1)

= u·(1−e−λ·1)

= u · (1 − e−λ)

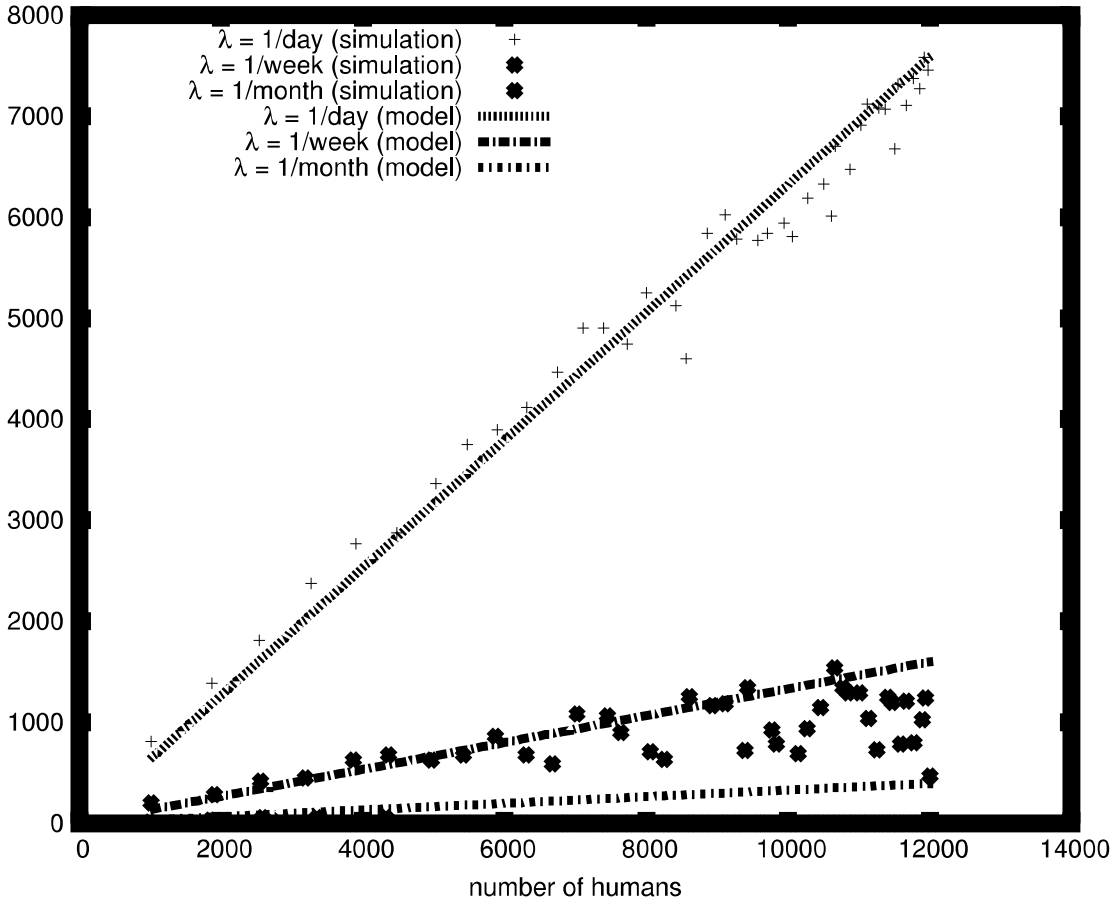
$\bar{d}$ = average disruptions per day

u = number of humans

PRPH= peer request per human

λ = new collaboration frequency

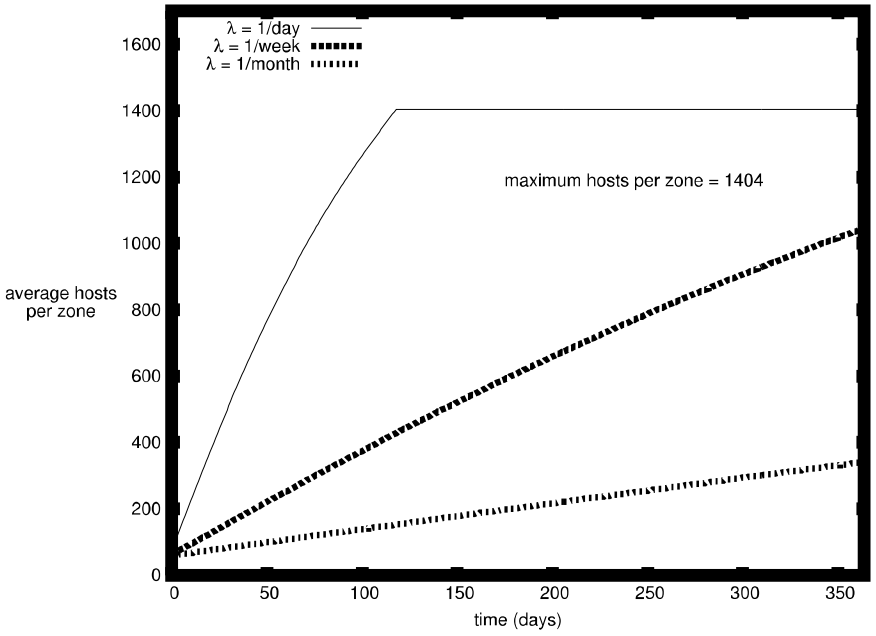
Steady State Metrics



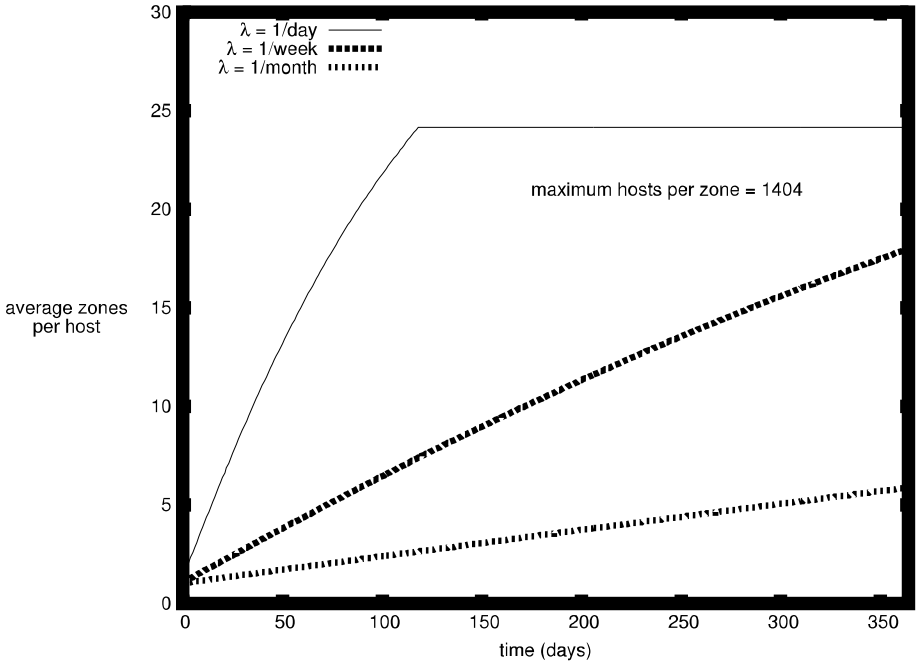
Number of disruptions versus number of humans

Algorithm's Performance

Transient Metrics



Average hosts per zone over time

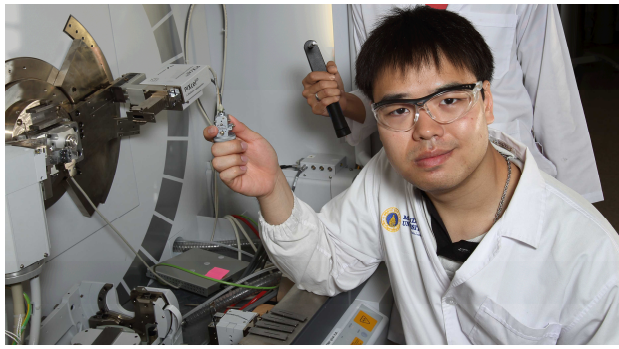


Average zones per host over time

Conclusions

- Dynamic zones are an important extension to previous research.
- Dynamic zones make attacks more costly and less feasible.
- In future work, we will pursue a number of refinements and enhancements to this study.
- Our future work also includes hierarchical zones





A Zoning Algorithm for Dynamic Cyber Zone Defense

Marci McBride [mmcbri@sandia.gov, (505)609-4303]
Rob Mitchell [rrmitch@sandia.gov]

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXXXP

Algorithm's Method

- The Results we obtained were via simulation.
- We obtained a list of the host names for all network devices for an enterprise.
- To generate each day's peer requests, we simulated humans forming new collaborations at some rate.
- We assumed this rate was guided by an exponential distribution because this distribution describes the time between events that occur continuously and independently at a constant average rate.