SAND2010-0986C

Ten Million and One Penguins
SAND 2010-xxxx

Ron Minnich

Sandia is a multiprogram laboratory operated by Sandia Corporation, a
Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia
National
Laboratories

The idea

- We're working to boot ten million machines
- We'd like to run a real botnet at scale and scale seems to be “huge”
- Of course, the numbers are open to argument, but …
- “A computer botnet is known to have breached almost 75,000 computers in 2,500 organizations around the world,” – last week
- Found almost by accident

Current situation: over 10M compromised in US

- No. 1: Zeus: 3.6 million
- No. 2: Koobface: 2.9 million
- No. 3: TidServ: 1.5 million
- No. 4: Trojan.Fakeavalert: 1.4 million
- No. 5: TR/Dldr.Agent.JKH: 1.2 million
- No. 6: Monkif: 520,000
- No. 7: Hamweq: 480,000
- No. 8: Swizzor: 370,000
- No. 9: Gammima: 230,000
- No. 10: Conficker: 210,000 <<= we thought this was bad!
- Source: http://www.networkworld.com/news/2009/072209-botnets.html

Sandia National Laboratories

How are botnets built?

- Typically “overnet” (nice writeup at wikipedia)
  - So-called because it is an overlay network
  - i.e. it has structure “overlaid” on the internet
- Using edonkey2 protocol
- The legal overnet taken down 2006
- Like that did any good, because:
- The illegal version out there, alive, and kicking
  - Just try to tell the RIAA that!
- If p2p is outlawed only outlaws will have p2p

Edonkey implemented kademlia protocol

- That's another long talk … and wikipedia does a better job than I can do
- Kademlia implements a Distributed Hash Table (DHT)
- Hash is 128 bits
- Nodes have a hash (i.e. 128-bit ID)
- Nodes contain information stored by hash as (key,value) pairs
- Hash uses XOR for “distance” metric

Kademlia network operations

- PING(hash) – what you expect
- STORE(hash, value)
- FIND_NODE(hash) – recipient of request returns set of nodes with least “distance”
  - For nodes, you want “close to”, because you already know yourself
- FIND_VALUE(hash) – return value of exact match of hash
  - For values, you want an exact match of course

DHT information values

- For talking to a node: (IP, port) – can be used to contact other nodes
- Otherwise, whatever you want
  - Movies
  - Songs
  - RIAA takedown notices
- And here's an interesting thought:
  - Executables
  - Command files
  - commands

Put it together

- You have a way to uniquely name a node with low probability of collision
- You have a distributed way to:
  - Find a node
  - Join the set of nodes
  - store information
  - query information
- So you've got a fault-tolerant, distributed, programming support environment

RIAA shut down the legal uses

- But it's all there for the bad guys
- And they use it
- Again, that's another very long talk but, as usual, wikipedia has great foundation article
- The statistics are overwhelming
- And kind of hard to verify: how do you really know, if every attempt to probe it is foiled?
- But it's real enough to scare researchers
    - Some are physically afraid!

Sandia National Laboratories

So what to do?

- One possibility is to apply High Performance Computing (HPC) resources to attempts to understand behavior
  - 180,000 core/30,000 node “Jaguar” at Oak Ridge
  - 20,000 core/5,000 node “Thunderbird” at Sandia
  - And all those little 10,000 core systems out there
- This idea has (in some cases) met with both skepticism and outright hostility
- I'll address two objections here

Sandia National Laboratories

“The guys who wrote this stuff didn't need a supercomputer so you don't”

- Recall that a lot of the NRE for Storm was done by researchers (kademlia) and .com's (overnet)
- Their scale up was done on the Internet:
  - By consent when it was legal (pre-RIAA-takedown)
  - By deception after it was illegal (by criminals)
- The Internet, as you know, has literally dozens (of millions) of nodes
- As a legal entity, Sandia would have some difficulty pursuing this approach :-)

“You can't simulate it”

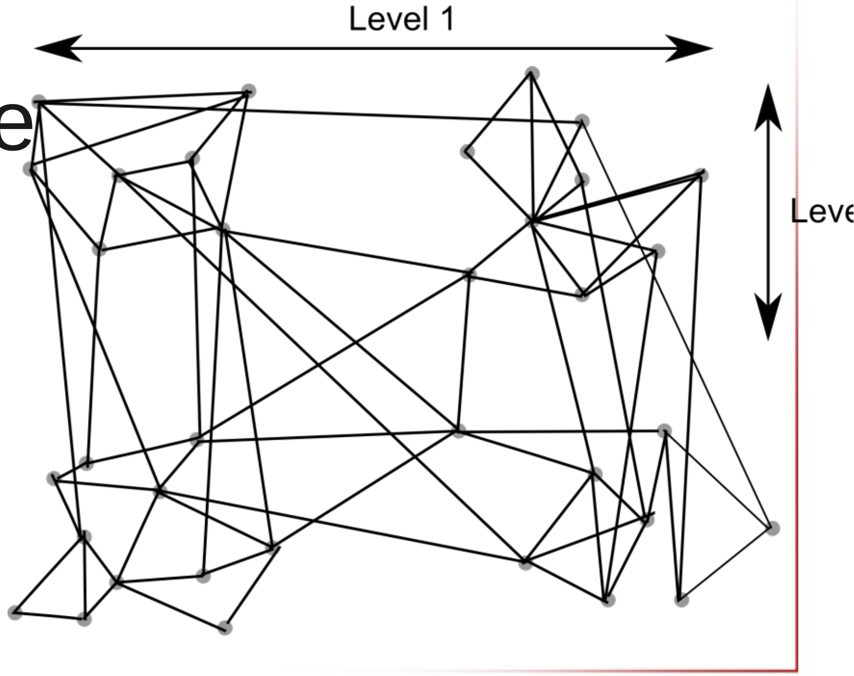
- If I had a dime for every time I've heard that claim over 30 years, well …
  - Every time we apply simulation to new areas, the [non-]experts in that area are revolting
- Let's just say that we have reason to believe we can do this
  - And, until we try, we won't know
- One STORM researcher: “Virtualization is the holy grail for us”
- So somebody thinks it's a good idea

Why run at large scale?

- We don't know on a large scale what
  - Denial of service
  - Exfiltration of data
  - Botnets
  - Virus transmission
- And other exploits look like
- *Can not be predicted by running 1000 or so*

Remember that 128-bit hash?

- “Distance” in the hash has no relation to distance in IP or geography
- So nodes “next to each other” in hash space can be anywhere
- Need to “populate” the space
- Or it just fragments
- Need at least 50K machines



Botnets exhibit “emergent behavior”

- So running 1,000 or so won't let you see its behavior at scale
- Only choice is to make the scale “big”

How to get to the scale we need

- We need to run a nation-scale network
  - We define this as 10 million nodes or more
  - Including routers
- Which, at this point, we can not afford
- We have an option however
- Given a large enough cluster, we can boot 10 million nodes on *virtual machines*
- Virtual machine software is a (not so) recent addition to Linux

Virtual Machines

- Kernels normally run in a privileged (a.k.a. Supervisor or Ring 0) CPU mode
- In the 1960s, IBM devised a means by which a *kernel* could run other kernels as a *program*
- IBM currently runs 7,000 VM's per machine
- Starting about 5 years ago, VM's came to Linux
  - There are now 7 different VM systems in Linux
  - Three are commercial
- Given a cluster of several thousand nodes, we can run 10 million Linux kernels via VM's

Sandia National Laboratories

What we're doing

- Use OneSIS cluster software (onesis.org)
  - Used to bring up 4600-node cluster (T-bird)
  - Relied on NFS root in earlier version
- Extend OneSIS with what we learned from Los Alamos Clustermatic (9grid.net/clustermatic)
  - Extremely light-weight, RAMdisk-based nodes
  - Can boot a node w/20M footprint
  - Compare to huge footprint of current cluster software such as Rocks, DOE CCE(TOSS(TriPOD))

Sandia National Laboratories

Result: extremely light nodes

- With lots of room for … lots of Virtual Machines
- On T-bird nodes, 250 are easy, x4600 nodes
- Modern nodes, 1000 are easy, x10K on RS
- So we've gone to 1M on T-bird
- And we hope to go to 10M on Red Storm
- Was it easy? No. Success once, failure once
  - How I hate IPMI ...
- But it can be done.

Plus new stuff

- Pushmon
- bproc2

Building a nation scale network on a cluster

- For each cluster node (“host”), start 250-1000 VM's (“guests”)
- Need very low-overhead control system
  - Want the time to go to the VM's, not the host
- Need very small memory footprint
  - Primary limit on number of guests is memory
- Need to be able to efficiently start programs on 100s of thousands of guests at a time

Once it is booted, monitor it

- What VM's are up?
- What are they doing?
- What packets are they sending?
- We have the tools
- But this scale is 1000x the current scale
- Time frequency is at least 10x current frequency

Once it is booted and monitored, attack it

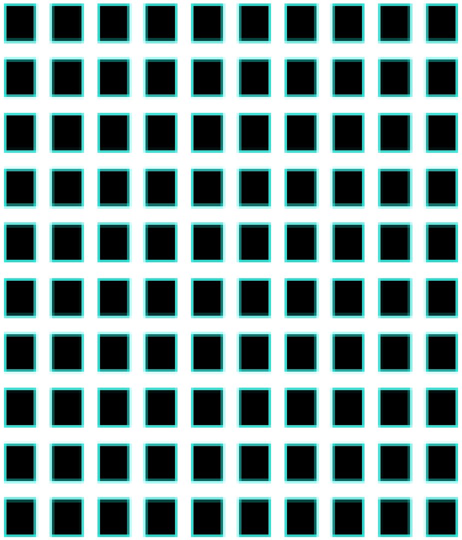
- i.e. find and run real malware
- This gets a bit tricky on your company network ..
- See if we can statistically characterize bad behavior from good
- And determine how much we need to monitor
  - Can we put probes at strategic points?
  - What is good enough?

Three-part program

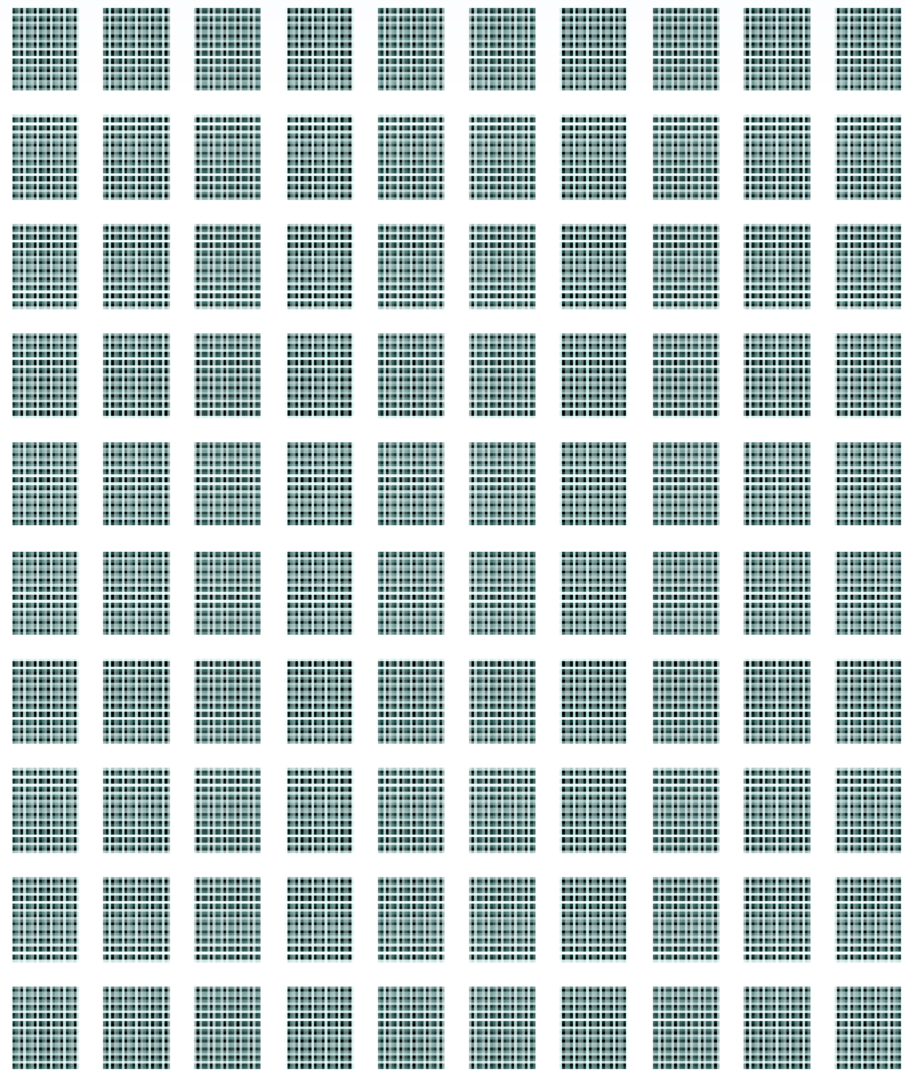
- Year 1: Boot 10M Linux kernels
  - Run programs on them (hard!)
- Year 2: Show that we can measure/control at this scale
  - Both “real-time” and “emulator time”
- Year 3: Show that we can emulate real cybersecurity events at this scale
  - DDOS against TCP stacks
  - Botnets
  - Worms

Sandia National Laboratories

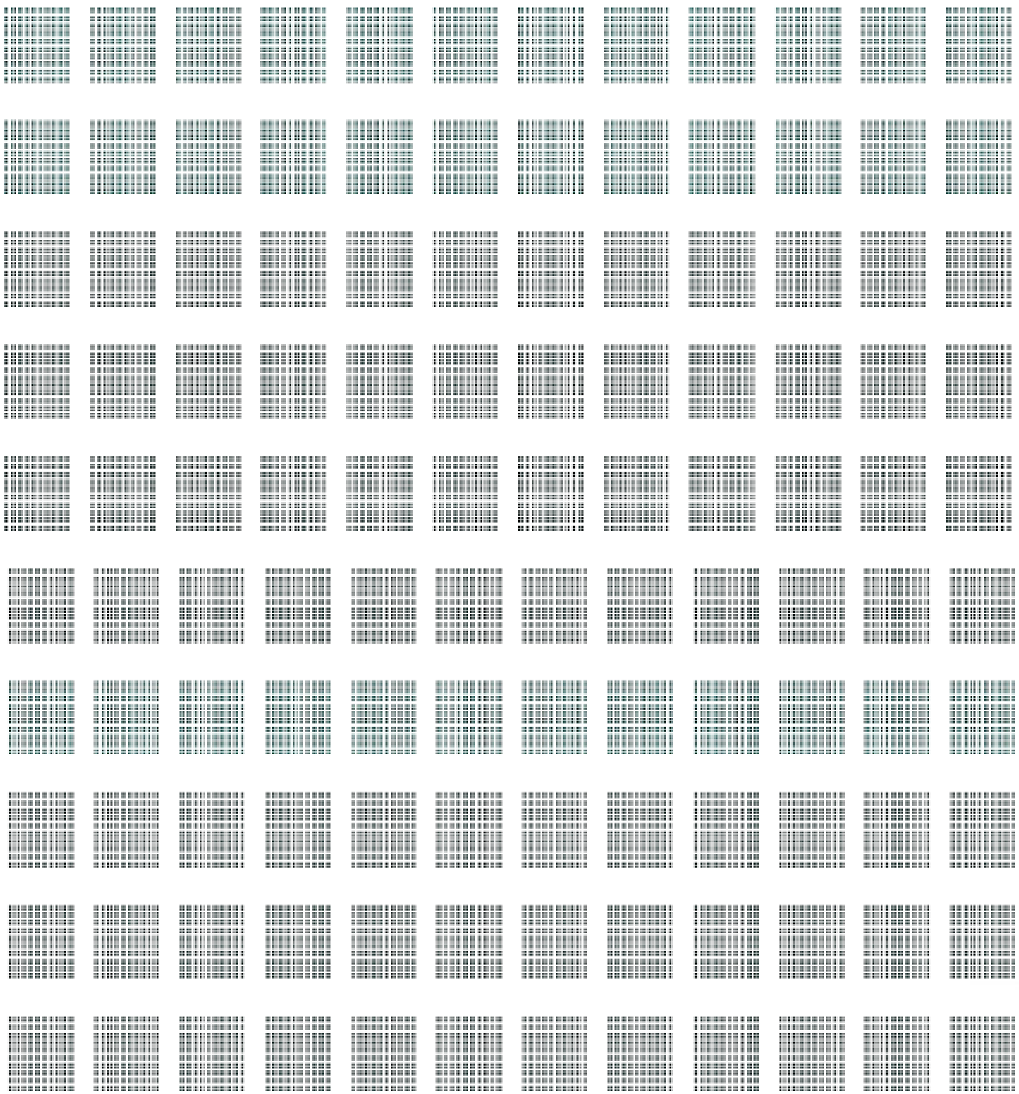
A note on scale: 100 nodes



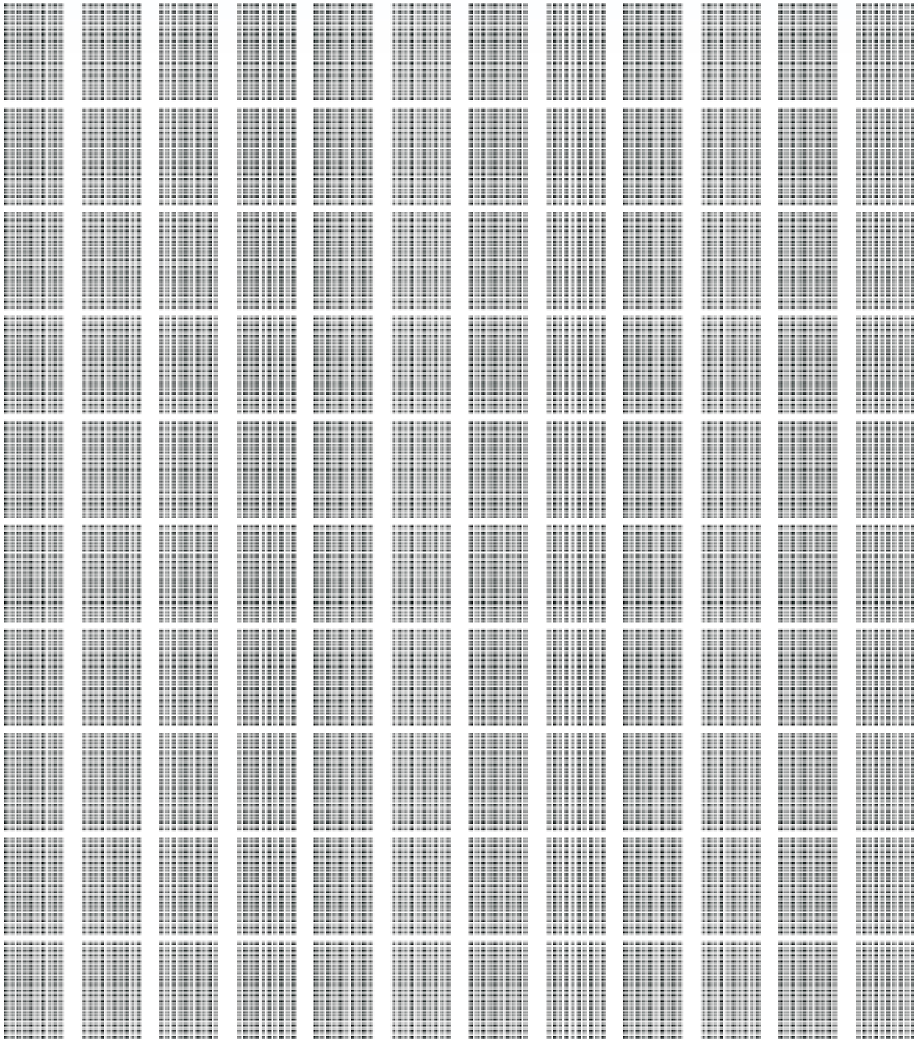
10,000 (supercomputer)



1M: 10x10 supercomputer we've run out of pixels ...



10M nodes: diffraction pattern



At 10M scale

- A DHCP file is at least 350 Mbytes
- Parsing /etc/hosts dominates startup time
- If all nodes talk to all nodes, kernel tables consume all of memory
- Even efforts to implement hierarchy get hard
  - Because in the end, using conventional tools, all the information has to go to/come from *somewhere*
- The tools we use today are designed for a small world
- This is a large world

Other issues (from experience at 50,000 VM's)

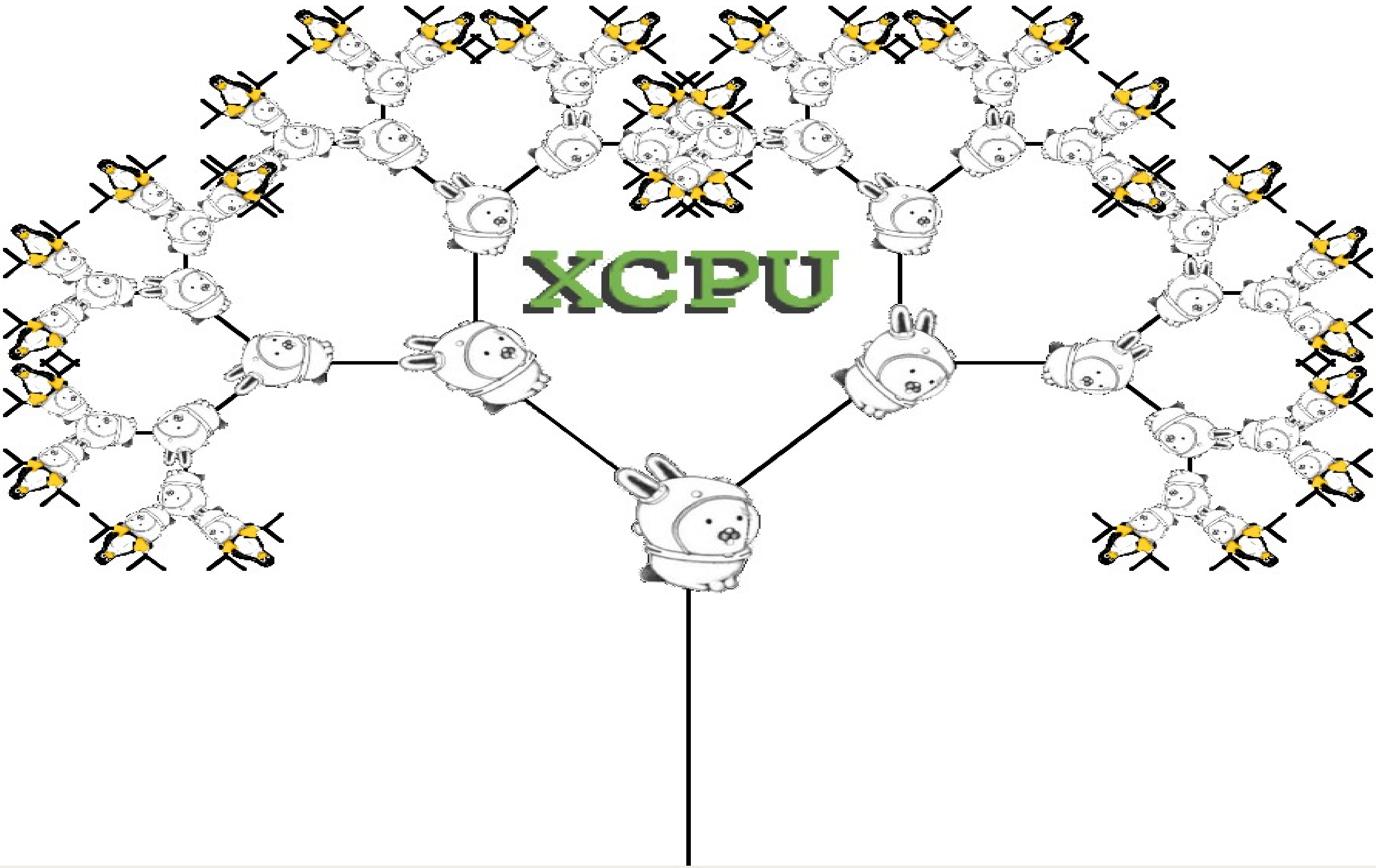
- Can not have global knowledge
- Nodes come and go with no warning
- Not possible to have all nodes booted at once
  - And how would you know if they were?
- Simple case: monitor nodes at 1 hz.
- 1 bit per node -> 1.2 Mbytes/second
- But we want more information

Sandia National Laboratories

To sum up

- Lots of machines
- No central configuration possible
- Status unknown/unknowable
- Firehose of data
- Hierarchy is fine but it has to be resilient
- *But we know the hackers and others do it*
  - Botnets of 100s of thousands are old news
  - One vendor had a 2M node botnet ca. 2001
- Might use botnet tools to run this network

Algorithmic self-configuration – for a logo



That logo was written by a program

- Other C code in distributed runtimes make similar decisions
  - Intermediate nodes convert from clients to servers
- Same self-configuring code used on Clustermatic clusters
  - No configuration files; node roles derived dynamically
- We need to take this model to the limit; algorithmic self-configuration for everything

Implementing a real network requires virtualized routers

- Any realistic emulation requires routing
- Sandia Sepia project demonstrated virtualized Cisco routers
- As nodes determine their roles, one role of a node or VM will be routing
- Will also require propagation of routing information using standard protocols

Year 2: Monitoring, analysis, control

- There are two parallel levels of monitoring
  - Real-time: managing the emulation
  - Emulator-time: managing the emulation data
- Number of real-time nodes is 1/250-1/1000th the number of emulation modes
- Emulation sample rate is 1/250-1/1000th the real time rate
- Example rates:
  - Real-time info: 1-10 MB/sec
  - 1/250th real-time: 16 MB/sec

Sandia National Laboratories

Year 2: Monitoring, analysis, control

Need enhanced capabilities for data collection and analysis

- Scalable, distributed, and highly-responsive
  - Gather and analyze large amounts of data on reasonable timescales (e.g. once per 10 “emulation” seconds)
- Robust
  - Infrastructure must function in face of continuous failure
  - Analyses must tolerate damaged/missing data
- Virtual Machines need to be extended to allow external monitoring
  - Need new high throughput monitoring interfaces
  - Code will be committed back to kernel

Year 3: Emulate and understand real cyber security events

Run a TCP-based denial of service from 10,000 nodes against the other 9,990,000

- Run a trivial worm
- Run an exfiltration application and try to dedicate patterns of data
- Run a botnet

Instrument the emulated components

- Analysis leading to attack detection

Response to attacks

- External commands to mitigate/contain attack
- Autonomous systems that can "heal"

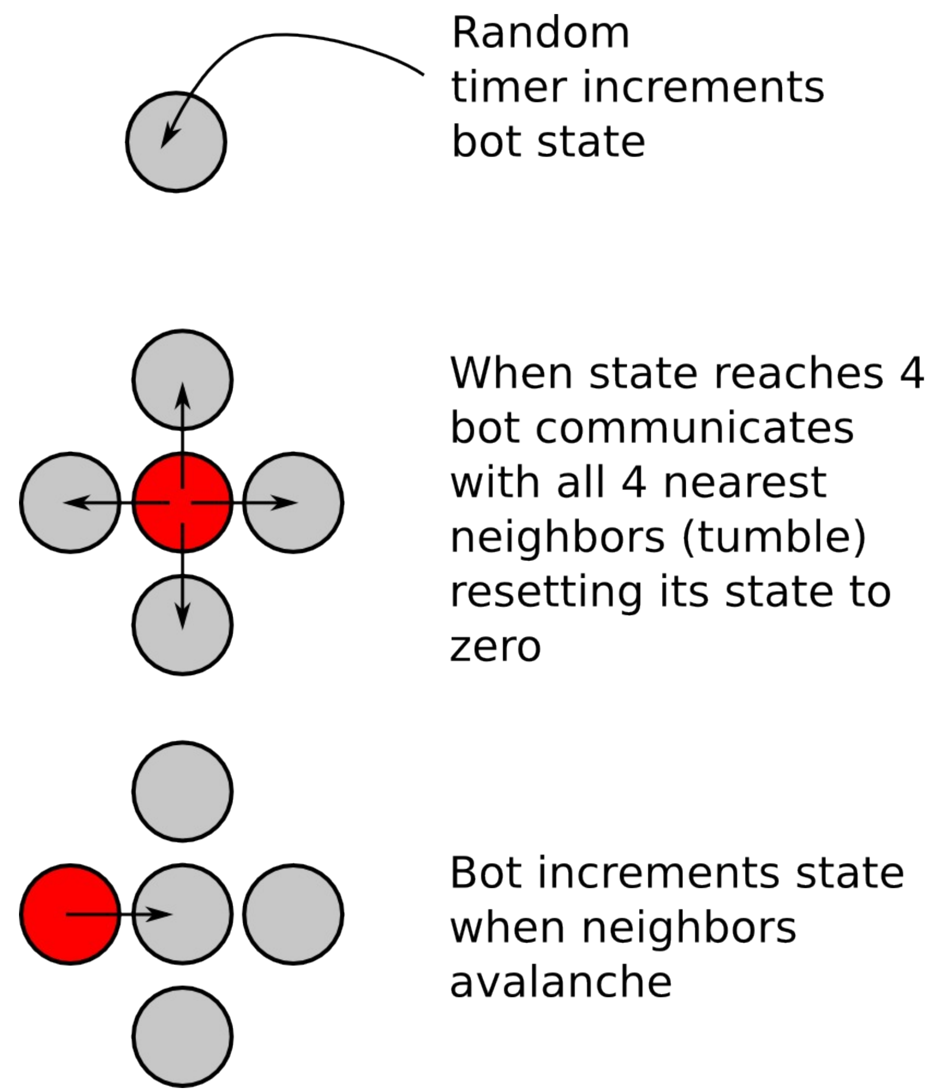
Sandia National Laboratories

Demonstration at SC 09

- A prototype botnet using >8000 machines on a computational cluster
- Data collected and the information presented in a movie where each pixel (!) is a machine (see next slide)
- Discover emergent behavior from realistic prototypes of highly organized botnets

"Sandbots" communicate via standard protocols (TCP/UDP) and obey simple rules

- Our simple "botnet" is based on a square lattice populated by bot nodes
- Simple rules determine when a bot communicates with its 4 neighbors
- Despite the model's simplicity, behavior at large scale is unexpected and rich
- Named "Sandbot" due to similarity to sandpile model of complexity theory



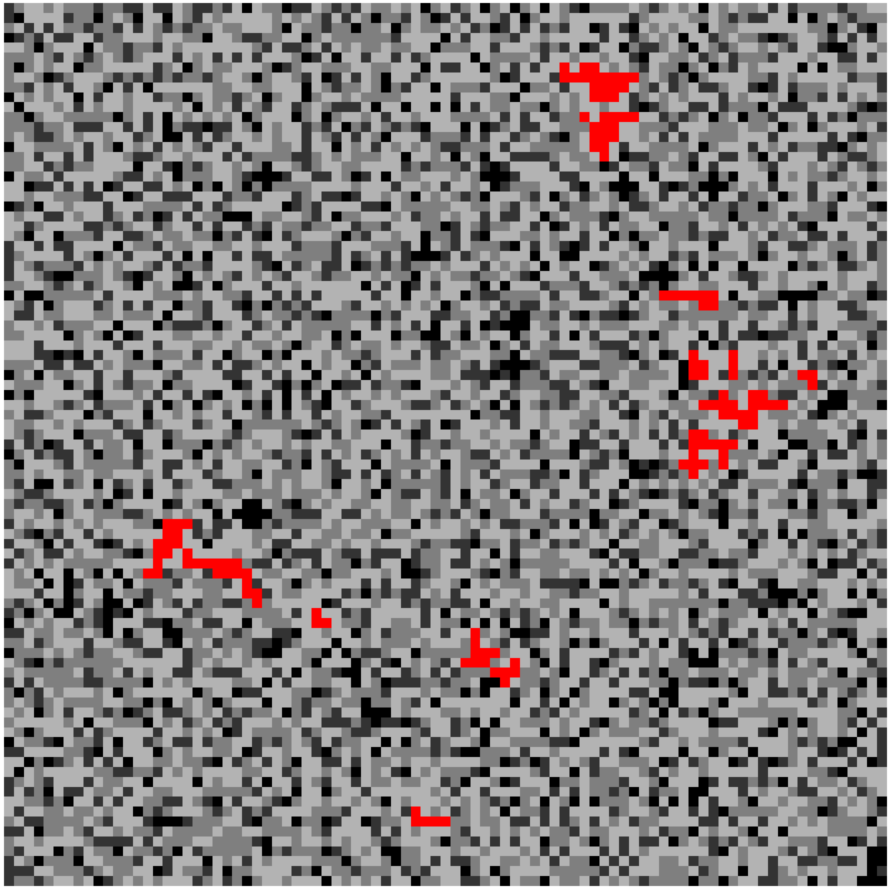
Sandia National Laboratories

Demonstration: Simplified prototype of large-scale, self-organized botnet

- Purpose: Demonstrate emergent behavior of bots on a large-scale network
  - Develop capability to emulate/simulate networks on a realistic scale
  - Discern instabilities and potential attacks that large-scale networks enable
- Platform: ~10^5 virtual machines on 10^2 physical nodes
  - Each VM runs a complete lightweight Linux operating system
  - Each VM is fully networked and uses the commodity protocols
- Bots arrayed in a square lattice, use a nearest neighbor gossip protocol
  - Requires 10^3–10^4 nodes to exhibit emergent behavior

Botnet reveals network "storms" ranging from smallest scales to the entire network

- "Small" example uses 8100 nodes so individual nodes can be discerned (have capability for 10^6 nodes)
- Red indicates an avalanche (i.e., a cascade of tumbles)
  - Smaller avalanches are more prevalent
  - Avalanche size will (rarely) span the entire grid
- Avalanches are chaotic but have well-defined signatures
  - Statistical features can be of use in intrusion detection



8100 machines running Sandbot: Each pixel is a single machine; network storms appear in red

Sandbot is prototypical of large-scale botnets

- Differences from real botnets
  - Organization: lattice not the fractal Kademlia network used by most botnets
  - Synchronization: VMs’ proximity—network latency needs to be better crafted to emulate reality
- Similarities to real botnets
  - Logical organization (i.e., lattice) completely independent of Internet topology
  - Uses standard TCP on standard network stack in a standard OS: timings such as jitter and software latency are identical
- Predictions
  - Aside from underlying representation, the features illustrated in this simulation are likely similar to real malware
  - Features predicted by simulation can be used for detection of malware in the wild

Sandia National Laboratories

Summary

- 10^6 has lots of challenges
- So it's not surprising that everything breaks
- Have upset every apple cart: configuration, monitor, control, and program execution
- We have a simple bot that shows emergent properties, complex behavior
- Also working on using real bots but this is a tricky