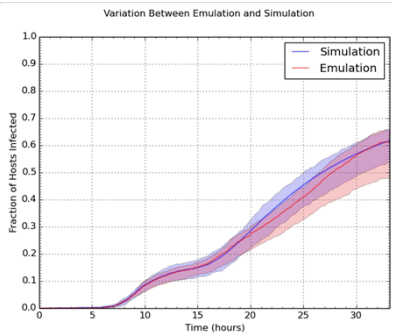
SAND2016-5882C

Exceptional service in the national interest

Sandia National Laboratories

FIREWHEEL



Cybersecurity Models

Complex Systems VVUQ Workshop 2016

Kasimir Gabert, Sandia National Labs

U.S. DEPARTMENT OF ENERGY
National Nuclear Security Administration

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

Outline

Sandia National Laboratories

- **The Problem Space**
- Modeling Techniques
- Example Problems
- Verification and Validation Thoughts

Scope of Cybersecurity

Sandia National Laboratories

- (Ambitious) Goal: eliminate surprise from our computers
- It is a large field, spanning the whole computer system space: from subtle software / hardware bugs to the motivation of cyber criminals to unintended radiation from physical devices

- Luckily, in order to be useful these systems are necessarily engineered to reduce surprise

Past (Surprising) Events

Sandia National Laboratories

- January 2010 Operation Aurora
  - A series of advanced attacks first reported by Google and aimed at dozens of American companies. These attacks took advantage of previously unknown vulnerabilities in Internet Explorer and both exfiltrated intellectual property and accessed email of targeted users
- April 2011 Amazon Outage
  - A configuration error during an upgrade disconnected a large number of storage machines; once the error was fixed they all automatically tried to replicate, causing a re-mirroring storm, thread starvation, and a large system failure
- April 2014 Heartbleed
  - A vulnerability in the OpenSSL library was discovered allowing for arbitrary remote memory to be read, including private keys. This impacted a massive number of servers on the Internet.

Purpose of Cybersecurity Models

Sandia National Laboratories

- Similar to cybersecurity: eliminate cyber surprise (using models!)
- We are asked to determine:
  - How a network or system design change might help or hurt (performance, security, usability)
  - How well a network can withstand or protect against an attack
  - Whether a new product will make a set of systems “more secure”
  - What an optimal deployment or design of a system change might be
- Some additional uses:
  - Help system developers prototype as they build
  - Help train individuals or teams
  - Dynamically explore a system with open-ended research questions

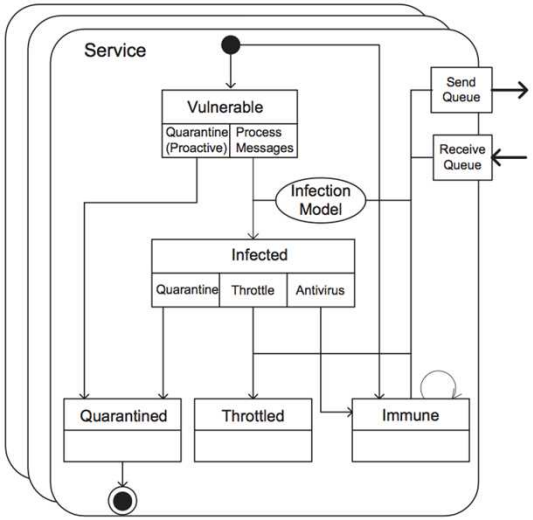
Outline

Sandia National Laboratories

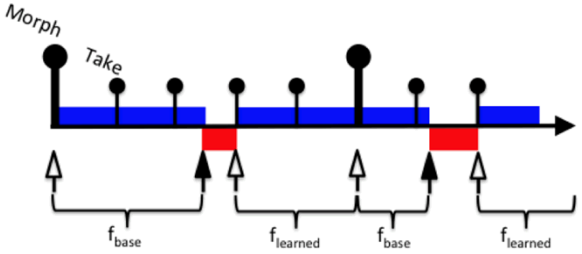
- The Problem Space
- **Modeling Techniques**
- Example Problems
- Verification and Validation Thoughts

Differential Equation/Agent-based/ Game Theoretic Models

- Critically, these seem to require a thorough understanding of assumptions and important parts of the system
  - Can this claim be strengthened through validation?



"Agent-based modeling of malware dynamics in heterogeneous environments", Bose et al. 2011



"Evaluating Moving Target Defense with PLADD", Jones et al. 2015
PLADD: Probabilistic Learning Attacker, Dynamic Defender

$$\frac{dI(t)}{dt} = \beta I'(t)S'(t) - \frac{dR(t)}{dt}$$

"Modeling Botnet Propagation Using Time Zones", Dagon et al. 2006

Emulation-based Models

Sandia National Laboratories

- Because we model computers using computers, the line between the “real world” and a “model” is blurry
  - A “real world” production virtual machine is a “model” element after being copied
- Emulation-based models (“Emulytics” at Sandia) take advantage of this blurry line by building models that consist mostly of virtual or physical machines running software pulled from the real world
- This approach can be augmented with simulation at a packet level or with the use of another (non-emulated) model

Model Assumptions / Parameters

Sandia National Laboratories

- Numerous assumptions – known/unknown, implicit/explicit
  - For every device in the network (from computers to network cables):
    - The operating system or firmware, from the broad version to every specific configuration option in every running service
    - The hardware the device is running on, all the way from rough specifications (amount of memory, speed of processors) down to the specific motherboard, all of the integrated circuits on it, and their initial RAM states
    - The software running on top of the system and all of its options and compilation / configuration settings
    - The users' interactions with the device
  - Networks can have thousands of devices in them
  - Model-specific software, for example a tool that mimics a user or a change that artificially speeds up a download to save time
- Unlike atoms, a different parameter can cause a state change

Running an Emulytics Model

Sandia National Laboratories

- The models are run (at clock rate) on computer systems
- We use virtualization platforms and numerous physical computers, each with their own complete set of assumptions and parameters
- A model may have significantly different output if it is oversubscribing the underlying hardware, experiences contention with other devices, or simply is not scheduled well across the physical cluster

- This adds a degree of non-determinism that does not seem to exist with many other models

Firewheel

Sandia National Laboratories

- A platform developed at Sandia that eases the process of building, running, and studying these models
- It is another large code base, filled with numerous assumptions, many parameters, and plenty of bugs

Outline

Sandia National Laboratories

- The Problem Space
- Modeling Techniques
- **Example Problems**
- Verification and Validation Thoughts

First Example: Shadscale

Sandia National Laboratories

- Project: Assist a customer in an annual tabletop cyber exercise by putting evidence behind discussed solutions
- Goal of the model: Given a concrete (invented) piece of destructive malware and a customer network, evaluate the different protections and mitigations the team comes up with

The Model

Sandia National Laboratories

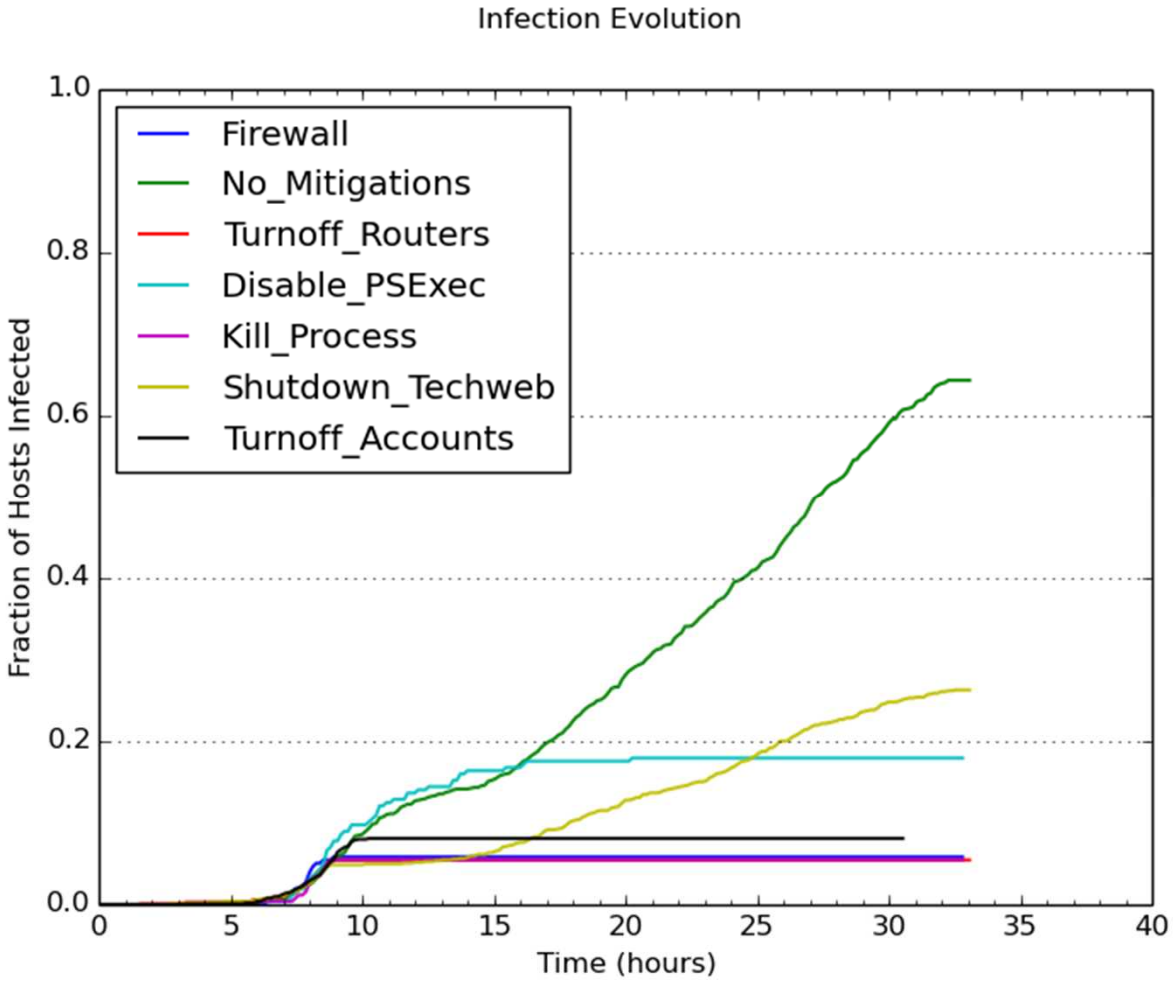
- Some considerations from the real world:
  - There are tens of thousands of computers on the network
  - Each has an operating system, physical hardware in some condition, various software installed, numerous real employee created data, etc.
  - There are many routers with unknown configurations, connected in a mostly unknown topology
  - There are thousands of employees
- We identified a single metric to evaluate mitigations with: Fraction of infected hosts over time
- To guide the model construction, we focused on 1) making sure we can output the metric and 2) making sure we can build each of the proposed mitigations

The Model Components

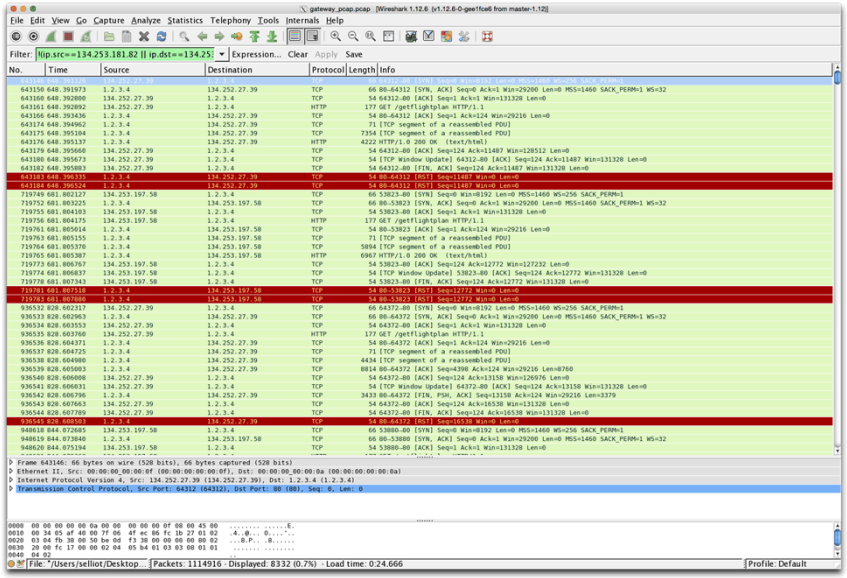
Sandia National Laboratories

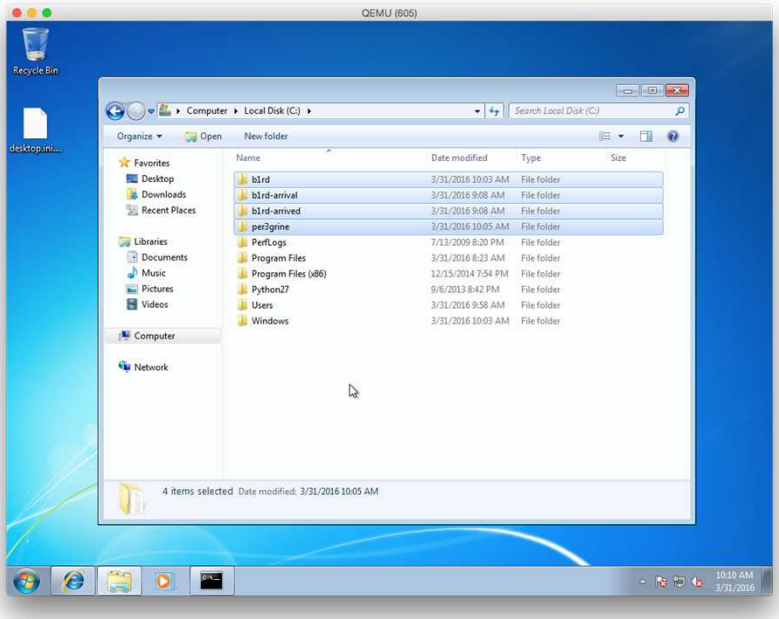
- Required by a mitigation:
  - Windows domain
  - Routers that can isolate subnets
  - Firewalls on network boundaries
- Required for the output:
  - Graphical interface automation in the Windows clients (the malware needs to be in the correct Windows security context)
  - Infection server replica
  - Required network infrastructure and servers to deliver and propagate malware
  - Simple user model (access per real log files, 20% click “Run”)

Results


Infection Evolution
Firewall
No_Mitigations
Turnoff_Routers
Disable_PSExec
Kill_Process
Shutdown_Techweb
Turnoff_Accounts
Fraction of Hosts Infected
Time (hours)
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
25
30
35
40

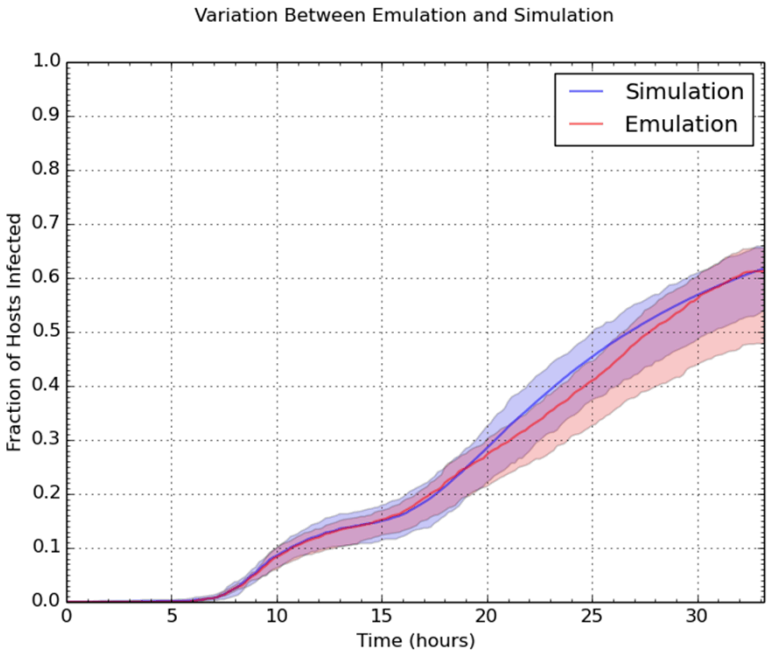
Shadscale Artifacts





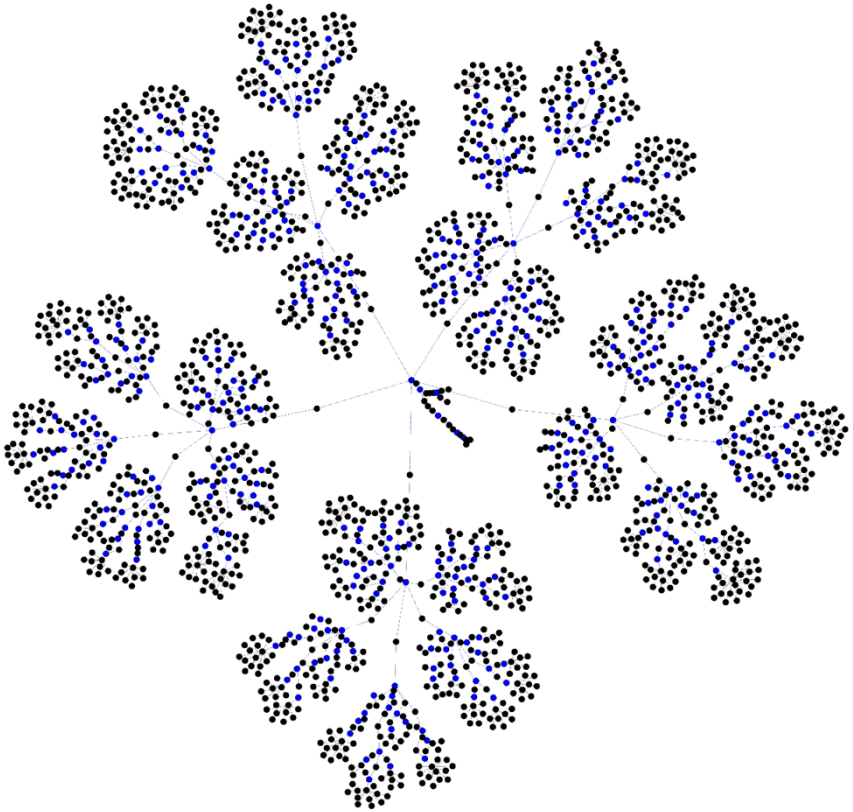
Sensitivity to Parameters

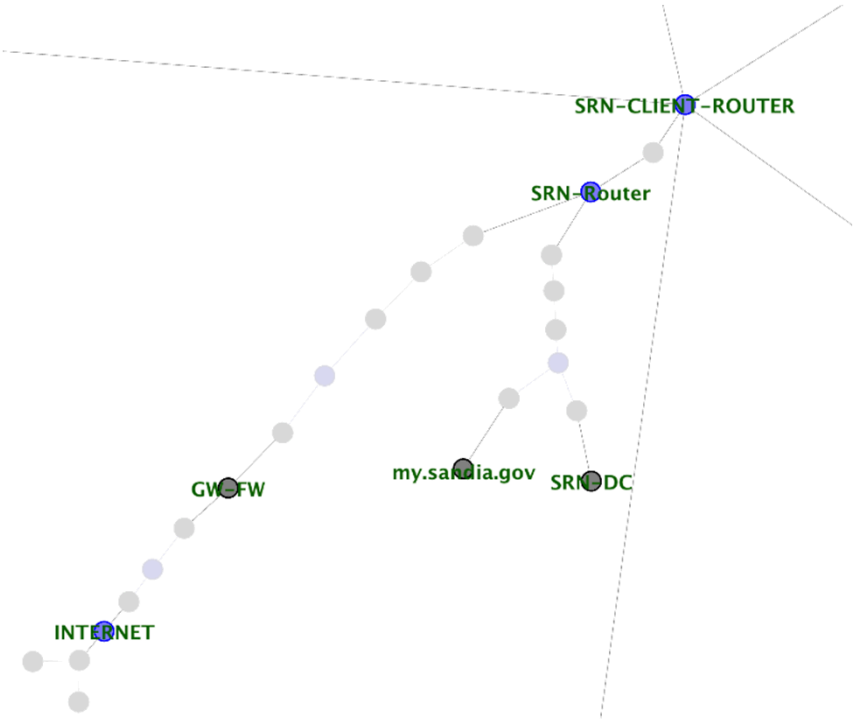
- A model this simple may be a good candidate for an initial validation pulling from traditional validation techniques
- The model (given that the artificial malware is so simple, even though it is peer-to-peer) seems to really only change given a different user model
- We built an agent-based model, erasing most parameters



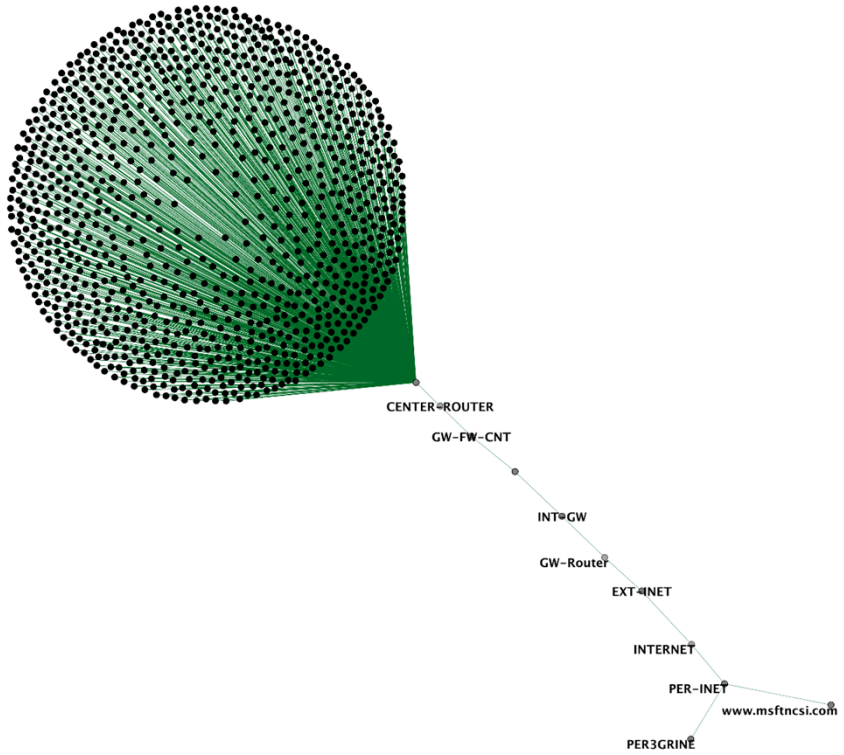
Topology Sensitivity

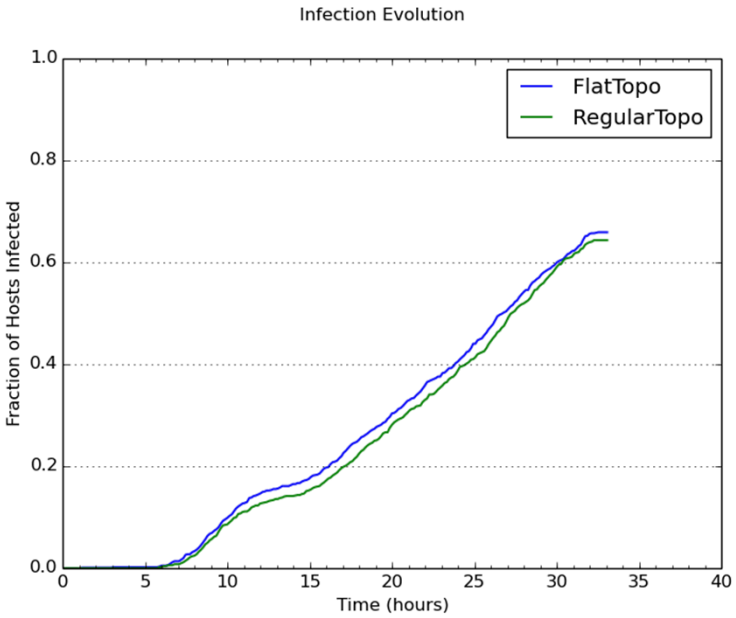
Sandia National Laboratories





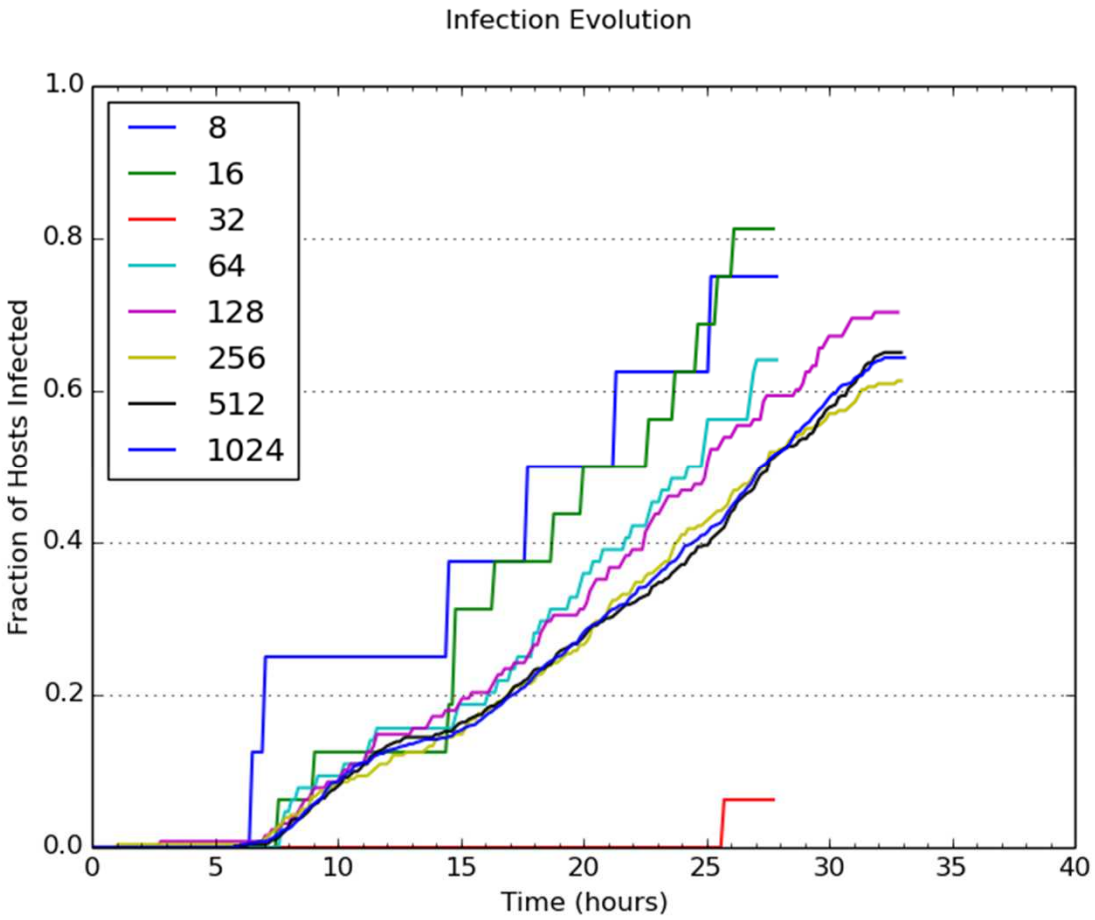
Topology Sensitivity





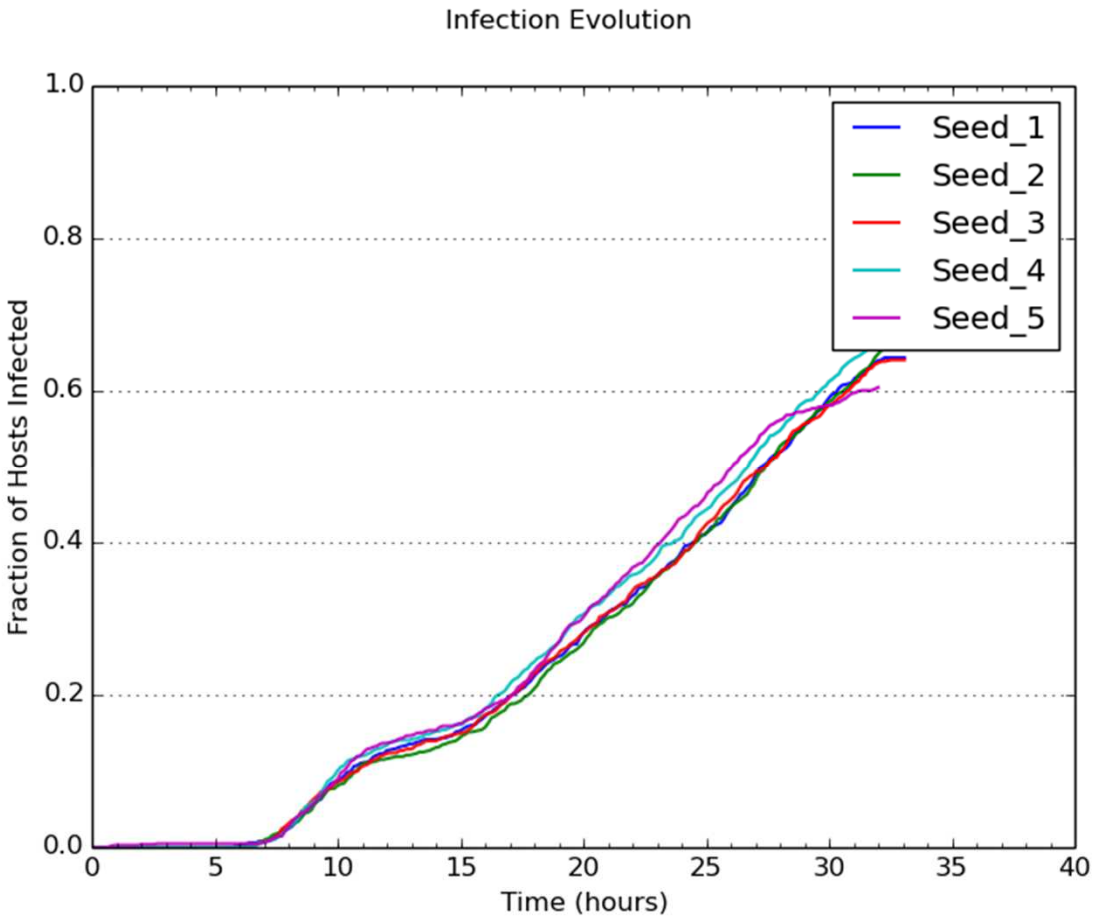
Topology Scale Sensitivity

Sandia National Laboratories



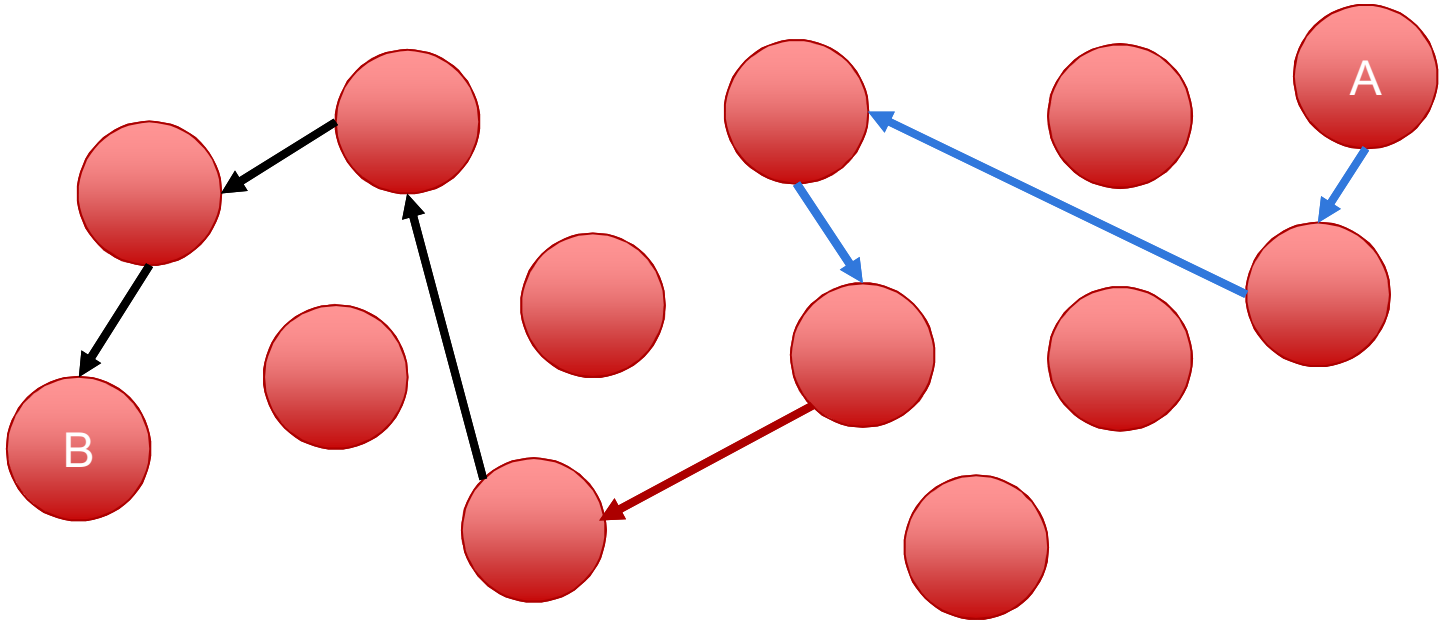
Topology Scale Sensitivity

Sandia National Laboratories

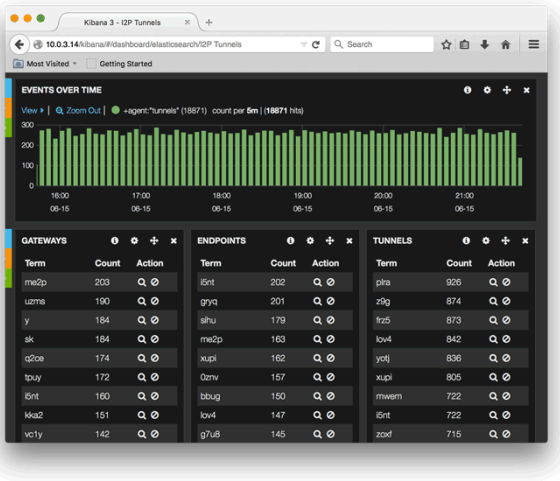


Second Example: I2P

- I2P (Invisible Internet Project) is a peer-to-peer overlay network designed to provide anonymous hosting
- Malware authors have moved communication servers into it—collaborating with Georgia Tech, we want to learn how to stop such malware



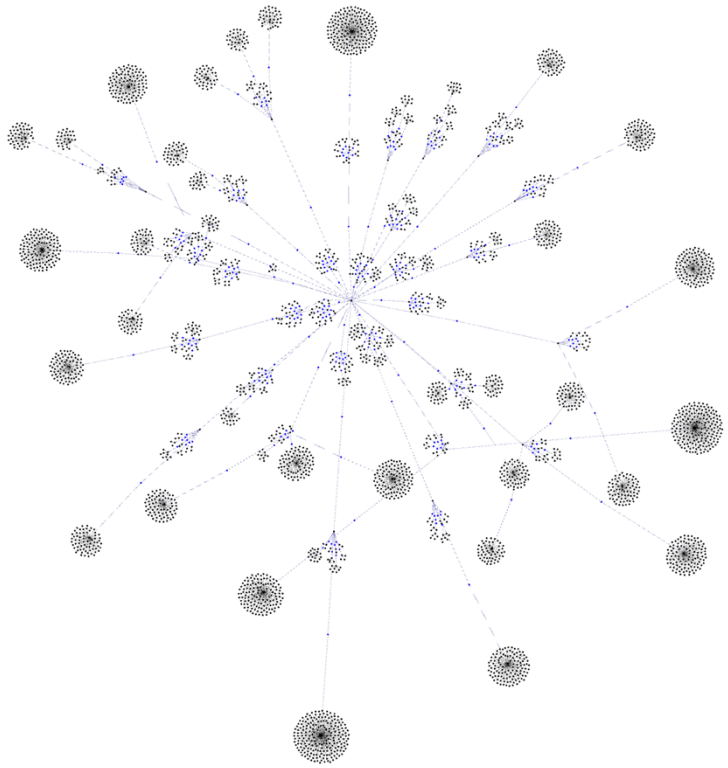
I2P Model Goals



1. Suppose we can temporarily turn off I2P traffic to an autonomous system on the Internet. How much would we have to do this to increase our tunnel ratios?
2. The I2P developers may be willing to update the code to ban the malware. Will all users who update isolate themselves from I2P?
3. It appears that a poisoned router entry could break I2P. (Testing this in the model showed that it is not the case.)
4. Given a small testing infrastructure, can we perform a larger population estimation by varying unknown model parameters?

I2P Model

- I2P software installed on Ubuntu 14.04 desktop clients
- I2P bootstrapping using our own keys
- Network from reversing a research paper
- User behavior: an infinite loop browsing one website



Are These Valid?

Sandia National Laboratories

- For Shadscale, how do we know that the mitigations would be as effective as the results say?
- For I2P, do we know that a poisoned router entry will do nothing?
- How can we know that our resulting population estimate bounds are reasonable?

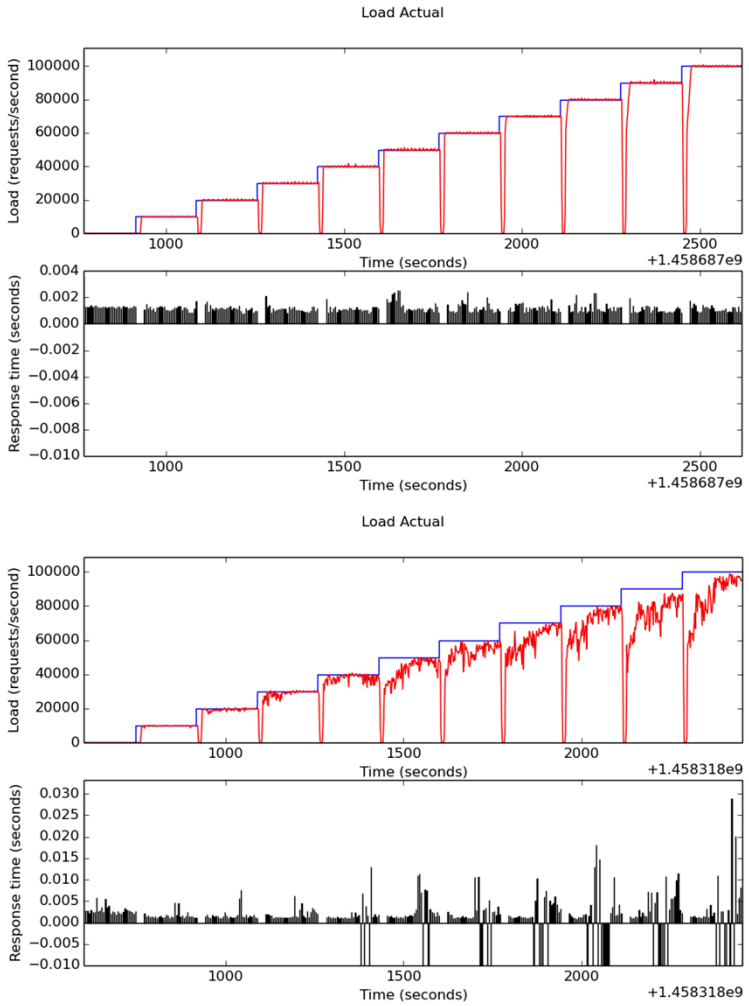
Outline

Sandia National Laboratories

- The Problem Space
- Modeling Techniques
- Example Problems
- **Verification and Validation Thoughts**

Visive Verdict LDRD

- Work in progress
- Identified problems: massive number of parameters and difficult metrics / quantities of interest
- We are building small laboratory models of key components – hierarchical may work



Possible Validation Distinguisher

Sandia National Laboratories

- As a general validation metric, use a Turing machine as a distinguisher between real and emulated
- D is a probabilistic Turing machine *distinguisher*, E is an oracle for the emulator, and R is an oracle for the real world
- Difficulty is in choosing the environment for D (the probability space and allowed accesses to the oracles)

- $Adv(D) = |\Pr[D(E) = 1] - \Pr[D(R) = 1]|$

Possible Verification

Sandia National Laboratories

- Software quality assurance seems to translate well
- For solution verification, can we compare against a “ground truth” oracle, one that responds with protocol descriptions or intended behavior (marketing literature?)?
- We may be able to build these verification checks into the models and allow them to run dynamically for each experiment

Questions / Discussion

Sandia National Laboratories