Mitigating Risk due to Undetected Malware Spread in Networks

Brian Thompson
James Morris-King
Hasan Cam

Motivation

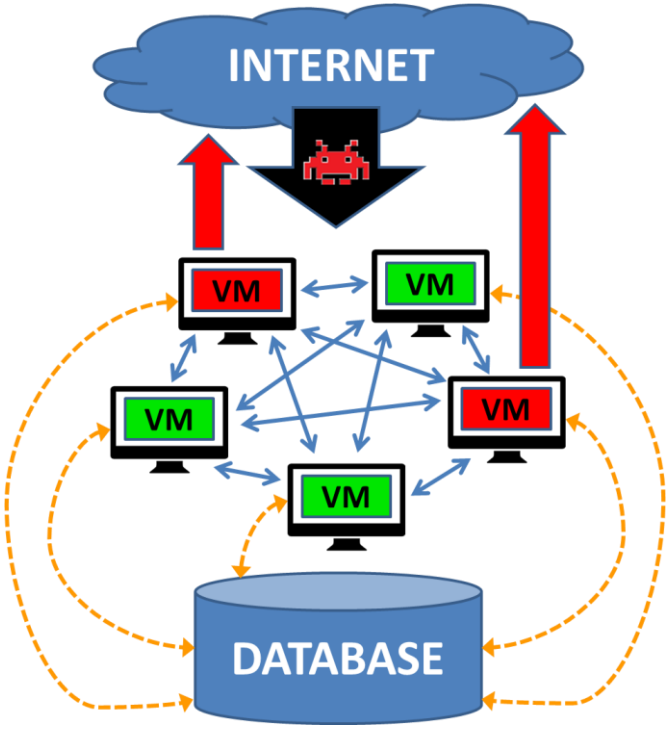
- The threat of cyber attack on businesses, governments, and organizations is real and dangerous
  - Propagating malware can sabotage critical infrastructure, disrupt services, and steal sensitive data
- Achieving 100% security is costly and impractical, if not impossible, in most real-world contexts
  - Duqu 2.0 (2015) – successfully infiltrated and spied on anti-virus company Kaspersky Lab's R&D technologies
  - DetoxRansome (2015) – hacked into security company BitDefender's network and leaked customer data
- Most existing approaches focus on detection and response; what to do when malware evades detection?

Problem Setup (1)

- Many organizations use a local area network (LAN) to provide high-speed internet to users (e.g. employees)
- Users log on in the morning, giving them access to company databases, and log off at the end of the day
- While logged on, users can communicate over the LAN, as well as connect to the internet for web browsing, email, etc.
- All activity is done through a virtual machine (VM), which is created at login and destroyed upon logoff

Problem Setup (2)

- Malware may enter the network when a user accesses the internet, for example via drive-by downloads while browsing the web or through email attachments.
- The malware then propagates undetected by bootstrapping existing communication, and exfiltrates sensitive data from machines it has compromised
- Because of the virtualization, malware is purged from a machine when the user logs off



Our Approach

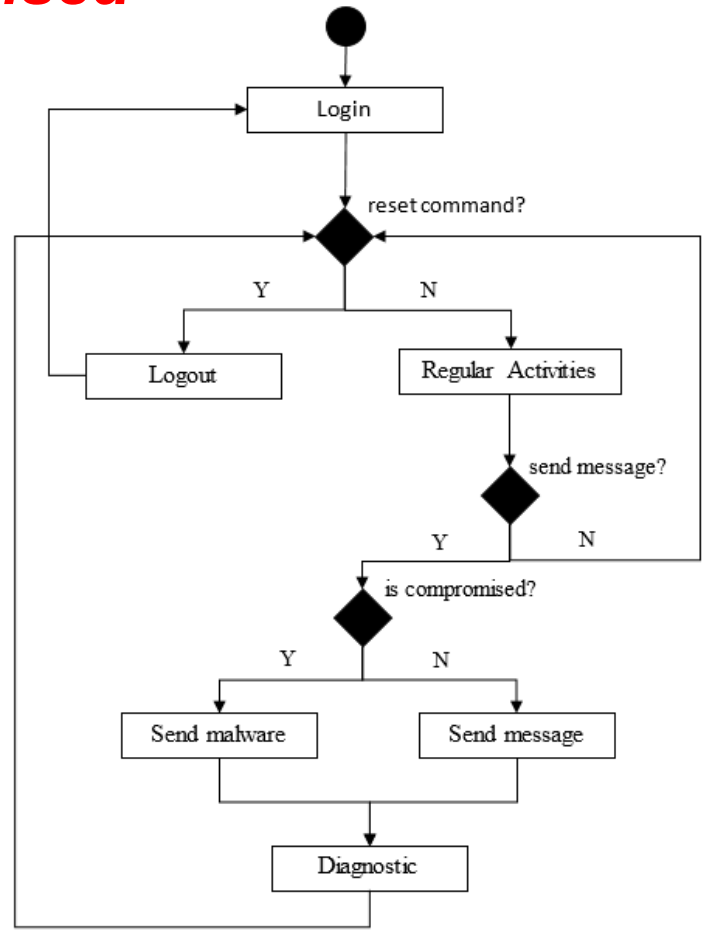
- We consider cyber defense strategies based on ***proactive reset***, in which users log off and log back on periodically throughout the day
- More frequent resets will result in higher security, but may also inhibit productivity, as users must interrupt their work in order for the reset to be performed
- We consider three policies that leverage data from activity logs to determine when users should perform a reset, in order to reduce risk of data exfiltration with minimal impact to productivity

Reset Policies

- **Time-limited policy**: Each user is expected to reset after a fixed amount of time has passed
- **Communication-limited policy**: Each user is expected to reset after receiving a fixed amount of communication from other users
- **Risk-flow policy**: A centralized system tracks all resets and all communication between users and maintains a risk score for each user, recommending a reset when the risk score exceeds a fixed threshold

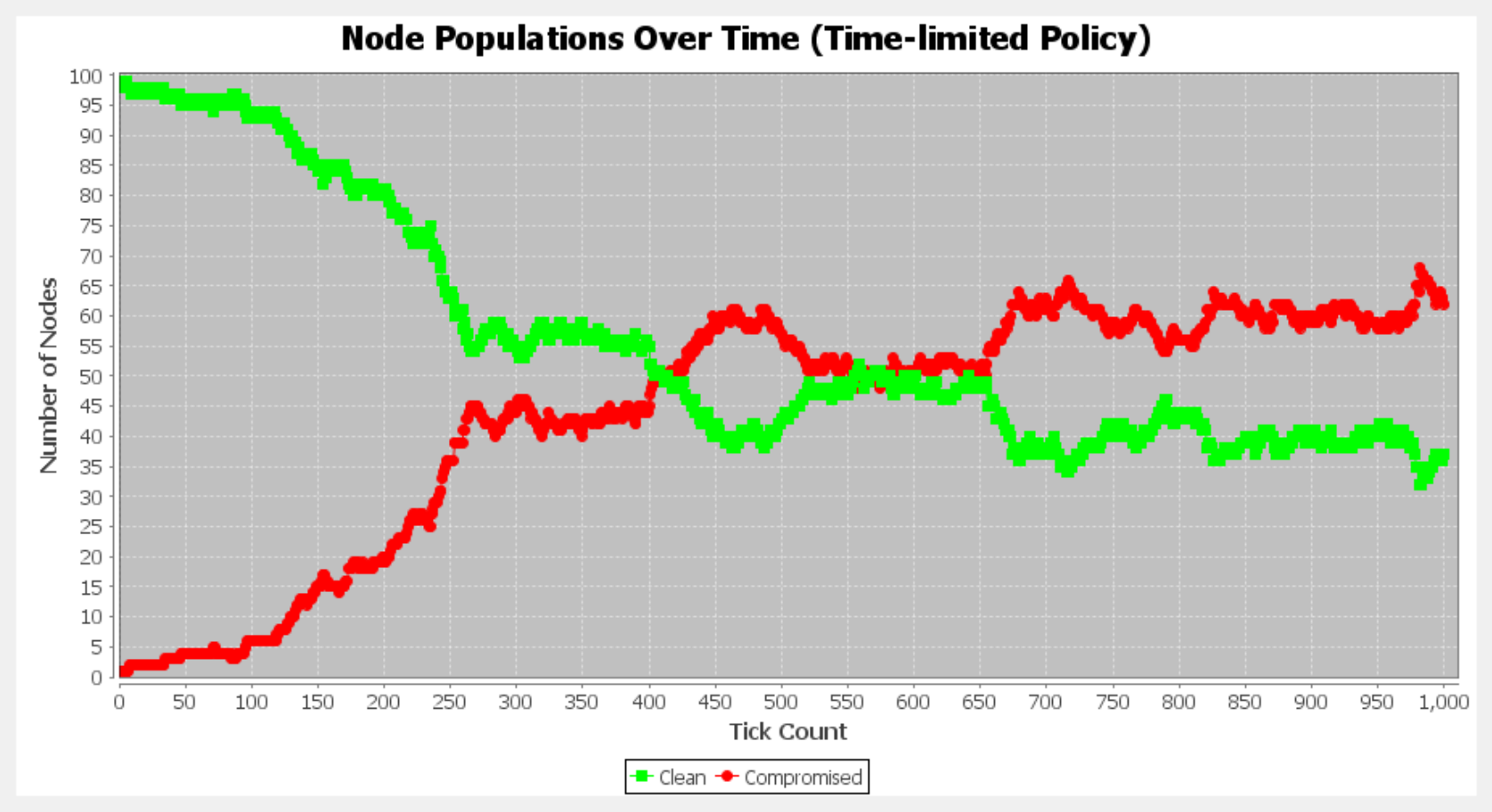
Agent-based Model

- Agents are ***clean*** or ***compromised***
- A clean agent can become compromised through external exposure or internal communication
- When a clean agent receives a message from a compromised agent, it becomes compromised
- The reset policy determines when agents should reset



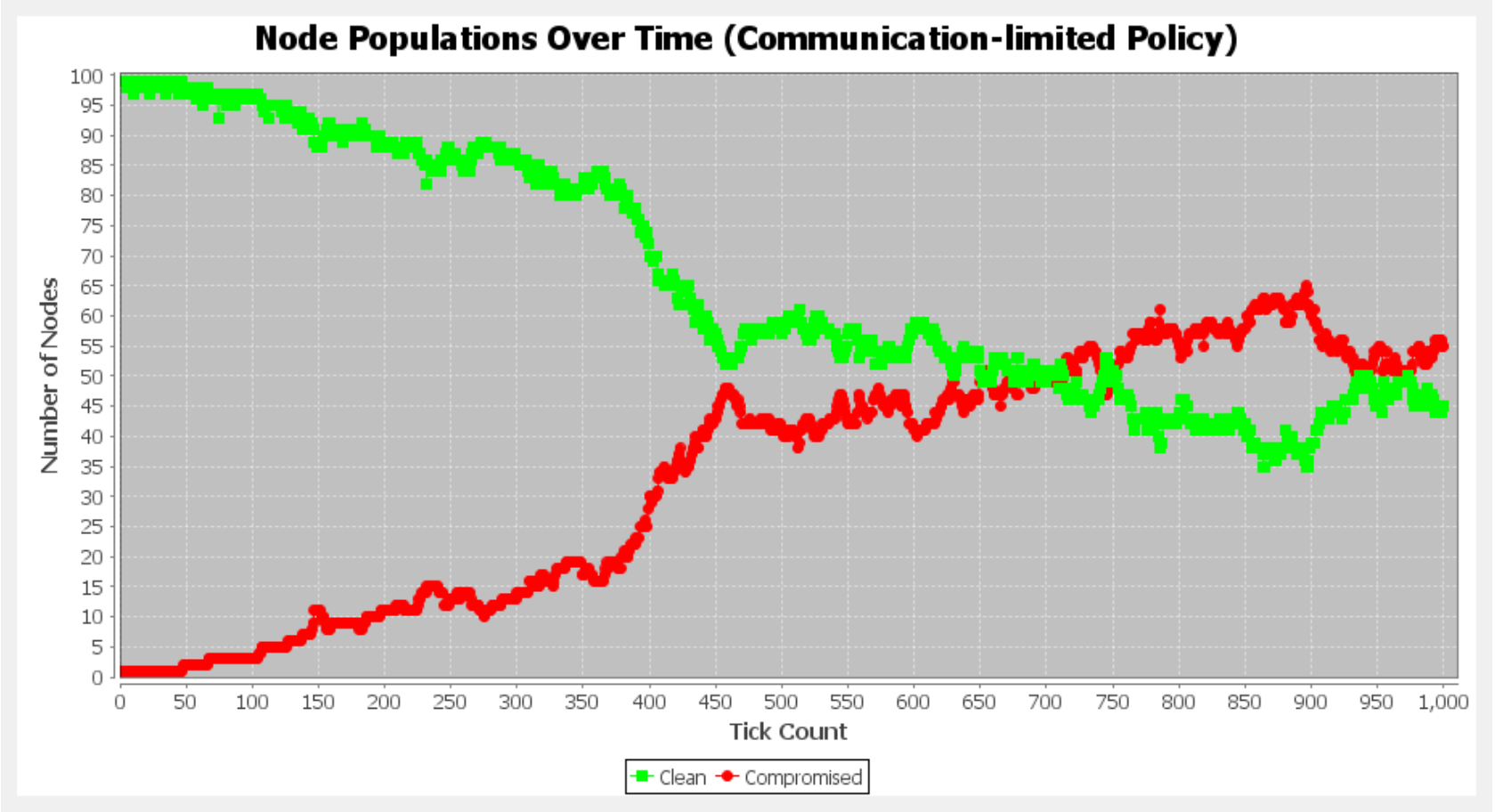
Simulation: Time-limited

send a message every 5 minutes, reset every 20 minutes,
single point of exposure at time t = 0



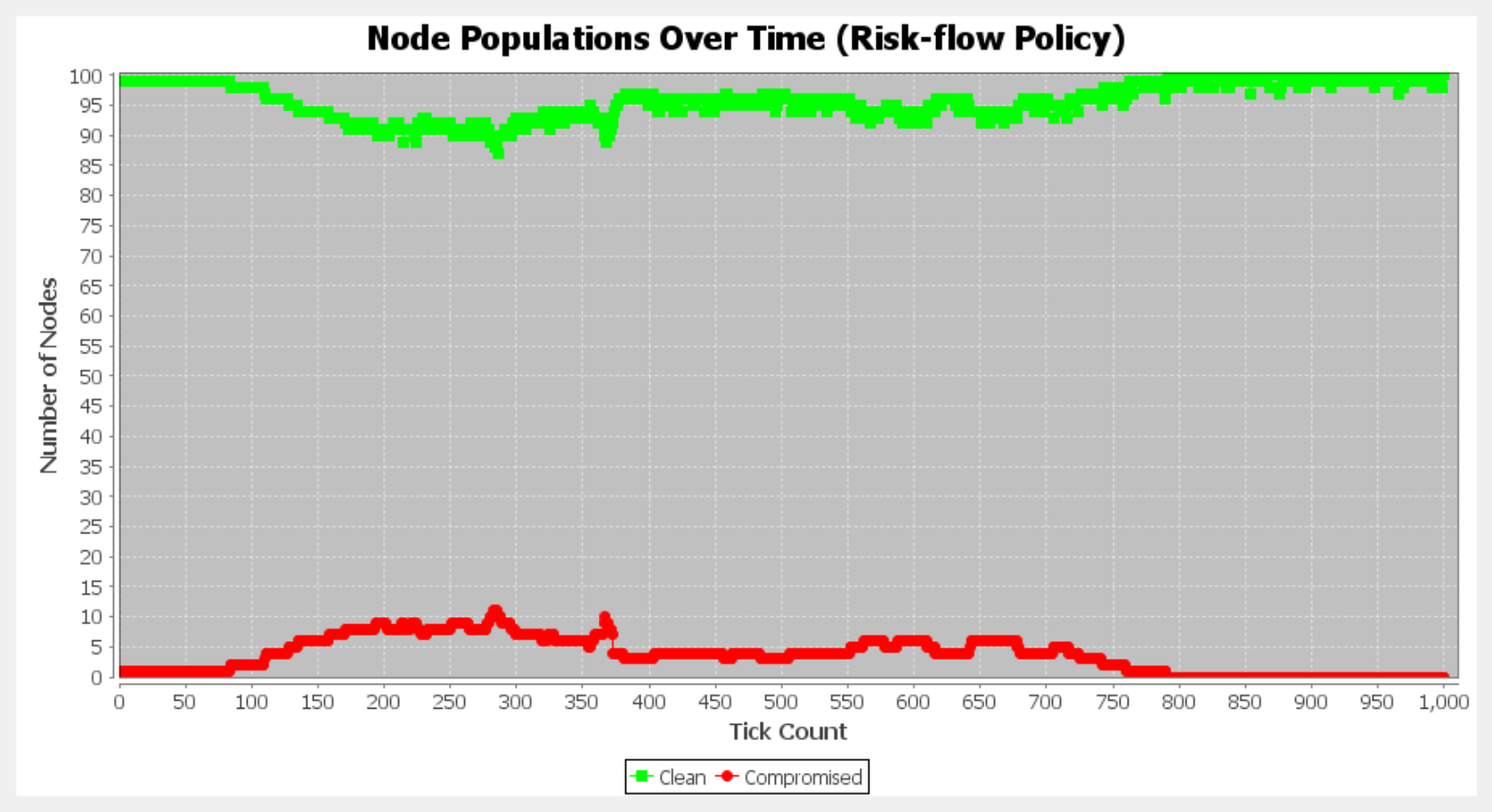
Simulation: Communication-limited

send a message every 5 minutes, reset every 4 received messages,
single point of exposure at time t = 0



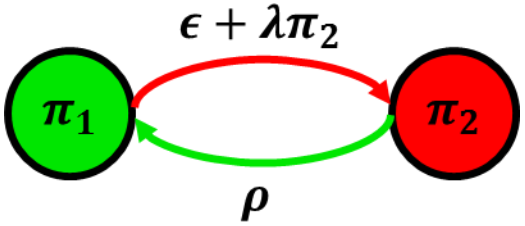
Simulation: Risk-flow

send a message every 5 minutes, reset on average every 20 minutes, single point of exposure at time t = 0



Baseline

- Compartmental stochastic model:
  - π_1 is the fraction of users with clean machines
  - π_2 is the fraction of users with compromised machines
  - ϵ is the external exposure rate for each machine
  - λ is the communication rate for each user
  - ρ is the reset rate for each user

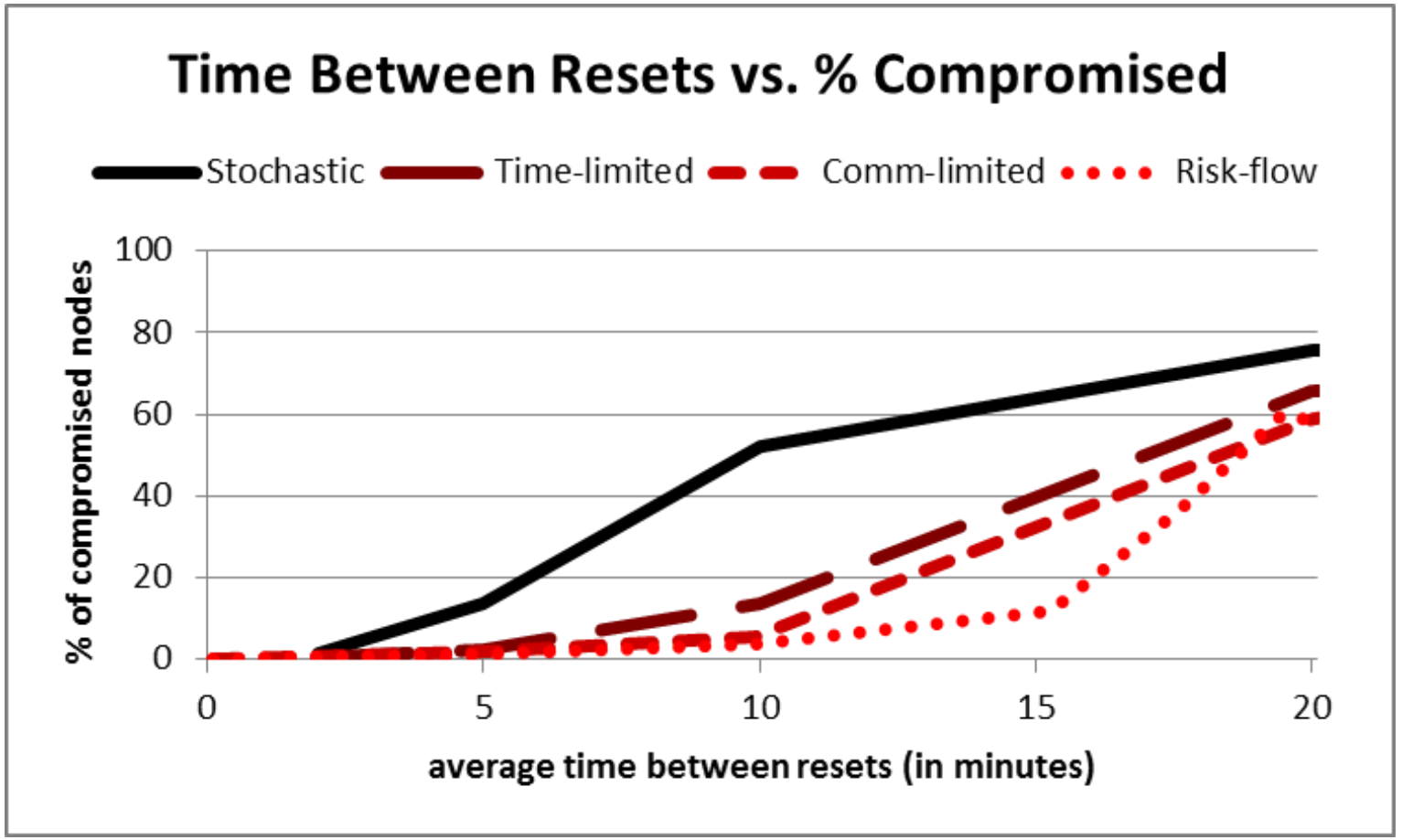


Equilibrium solution:

$$\pi_1 = \frac{\epsilon + \lambda + \rho - \sqrt{(\epsilon + \lambda + \rho)^2 - 4\lambda\rho}}{2\lambda}$$

Comparison of Policies

send a message every 5 minutes, reset on average every 20 minutes, exposure to malware from external sources on average every 4 hours



Summary of Contributions

- A proactive approach to cybersecurity that limits the spread of undetected malware through periodic resets
- Three policies to determine when users should reset based on data from activity logs
- Through agent-based modeling and simulation, we found that the risk-based policy leads to reduced data exfiltration compared to the other policies, for the same impact to productivity

Future Work

- Considering arbitrary network topology, or one based on geospatial proximity, instead of assuming all-pairs reachability
- Incorporating a mobility model
- Modeling the simultaneous propagation of different types of malware
- Considering a heterogeneous network where users have different communication rates, rates of internet usage, or database access rights

Questions?

Brian Thompson

bthompso8784@gmail.com

http://pidancer.com