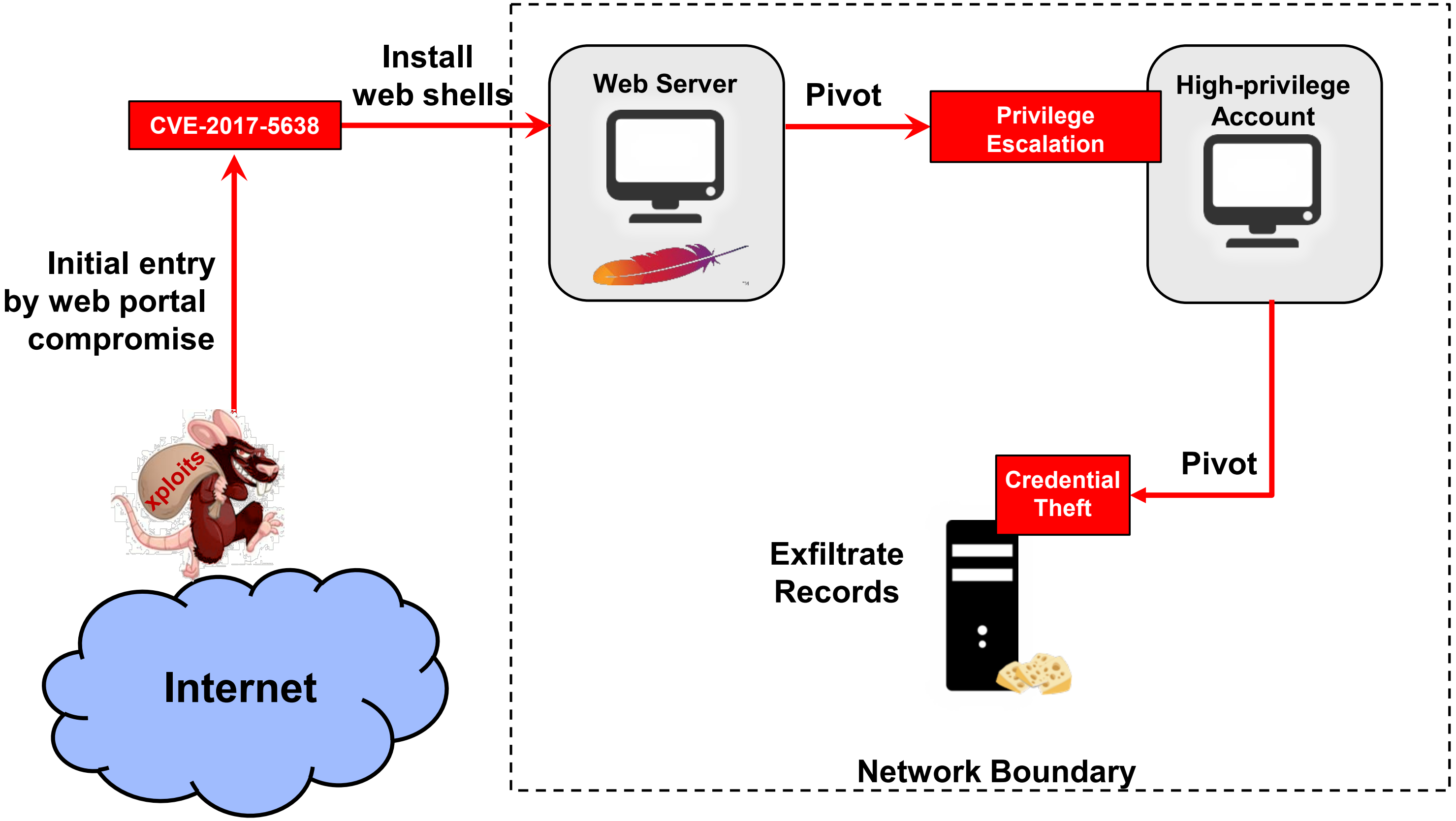
Proactive Cyber Situation Awareness via High Performance Computing

Allan Wollaber

September 26, 2019

LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Lateral Movement in the Equifax Attack



- Equifax observed suspicious traffic on 7/29/2017
  - Attacker had been in place since at least 5/13/2017
  - Exploited vulnerability to establish foothold
  - Spread laterally via allowed network communications
- Settlement cost: $700M
- Attack duration: at least three months
- This event is not unique
- Defenders are drowning in alerts

Is there a way to quantitatively enhance security in networks?

Approach In a Nutshell

- To provide cyber situational awareness (SA), the pythia prototype:
  - Learns the vulnerability arrival rates of software services
  - Learns their exploitation likelihood with an attacker model
  - Displays the extant vulnerabilities, hosts, and services in each network segment

- pythia can then project risk and trade-off alternative courses of action
  - Attacker/defender actions are simulated via a Monte Carlo model to project risk
  - A genetic algorithm (GA) explores many alternatives to recommend a more secure network segmentation
  - HPC makes this operationally relevant in a real-time setting

pythia enhances cyber SA and provides decision support

Vulnerabilities and Exploits 101

Vulnerabilities

- Recorded in the National Vulnerability Database (NVD) and maintained by NIST
  - Each vulnerability assigned a Common Vulnerabilities and Exposures (CVE) number
  - Over 120,000 vulnerabilities present
- Each is rated by the Common Vulnerability Scoring System (CVSS)
- Every publically known vulnerability is entered into the NVD and assigned a CVSS score (1-10)

Exploits

- Exploit: a piece of code which actually uses *(exploits)* a vulnerability to seize control of software or a host in a network
- Found in benign, e.g. Metasploit, and malign environments, e.g. the Dark Web
  - Example public exploit database that maps vulnerabilities to exploits at exploit-db.com
- Most vulnerabilities (>90%) in the NVD have no publically known exploits

Background

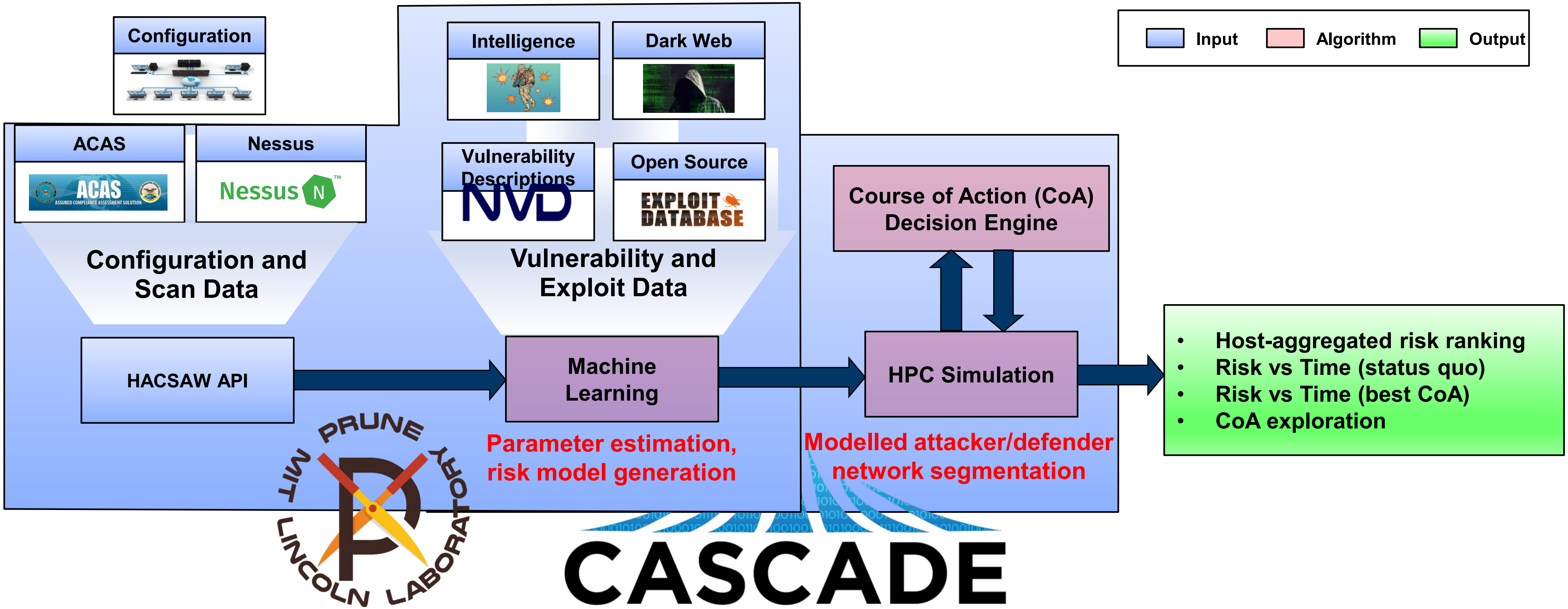
- **Vulnerability scanners help defenders comprehend the current situation, providing “level 2” Cyber SA**
- **Limited resources make it difficult to prioritize remediation response**
- **Attacker-agnostic risk scoring also under- and over-estimates current risk**
- **Modeling capabilities for testing and evaluation exist, but not in an operational capacity*. Our approach builds on that foundation.**

Name	IP	Score	Vulnerabilities	Exploit Available
HOST_NAME	127.0.0.1	1	91	Yes
HOST_NAME	127.0.0.1	0.94	26	Yes
HOST_NAME	127.0.0.1	0.9	22	Yes
HOST_NAME	127.0.0.1	0.89	21	Yes
HOST_NAME	127.0.0.1	0.88	20	Yes
HOST_NAME	127.0.0.1	0.87	19	Yes
HOST_NAME	127.0.0.1	0.87	19	Yes
HOST_NAME	127.0.0.1	0.83	17	Yes
HOST_NAME	127.0.0.1	0.83	17	Yes

Notional “wall of red” showing risky hosts.

*Wagner, Neal, et al. “Automatic Generation of Cyber Architectures Optimized for Security, Cost, and Mission Performance: A Nature-inspired Approach ” Advances in Nature-inspired Computing and Applications, 2018.

pythia Architecture Overview

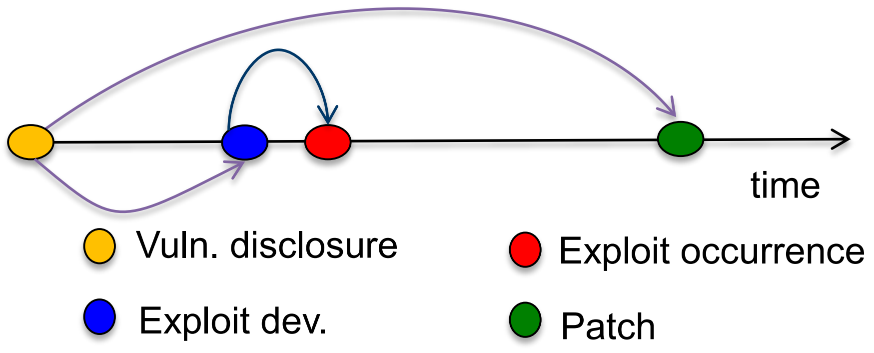


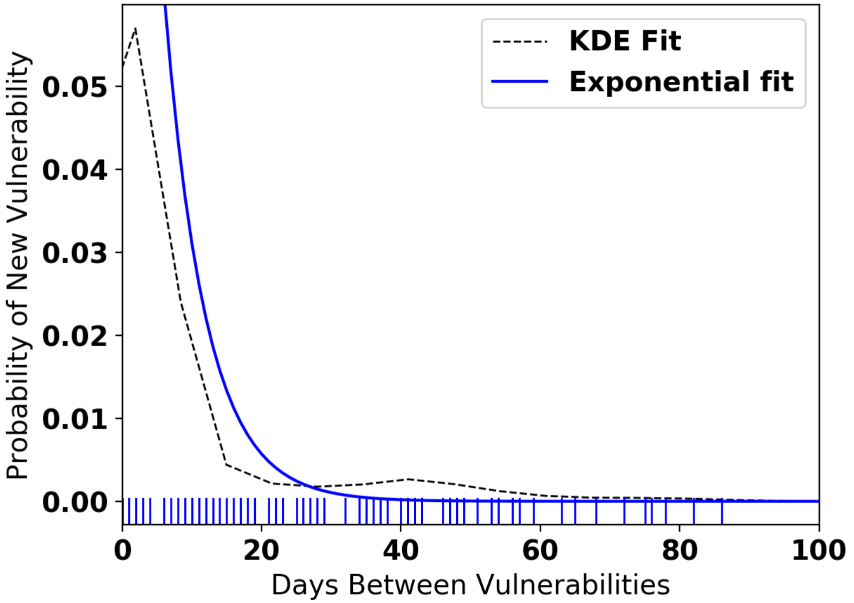
Pythia leverages HACSAW data, machine learning, and HPC simulation for Cyber SA

CASCADE: Cyber Adversarial SCenario modeling and Artificial intelligence Decision Engine
CoA: Course of Action

Vulnerability Lifecycle

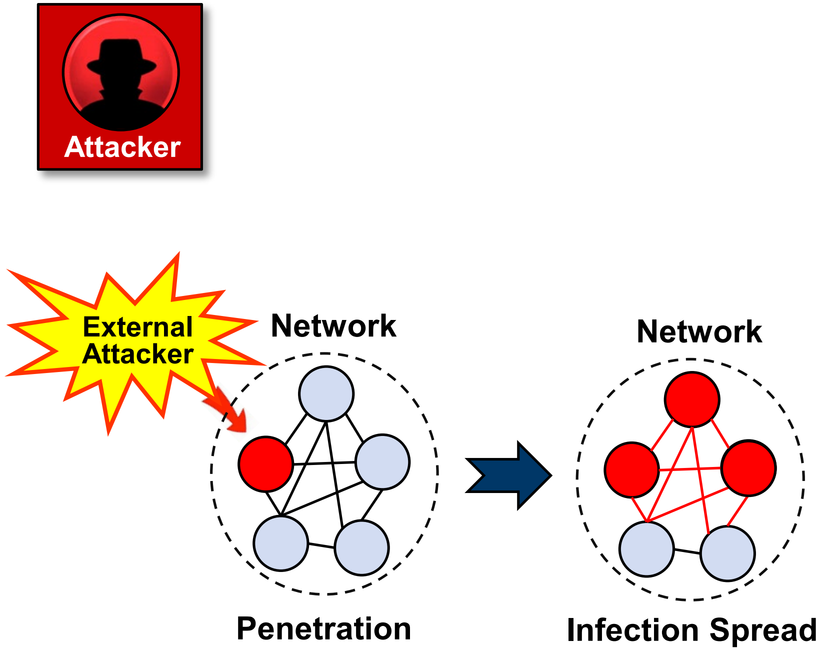
- **Software services nominally begin "clean"**
  - **Eventually, a vulnerability is found and disclosed**
  - **Attackers can develop and launch exploits as soon as they know of the vulnerability**
  - **Vendors can supply a patch to remove the vulnerability**
- **The patch may occur before or after the exploit is developed**
- **Each software service has a learnable history of vulnerabilities**

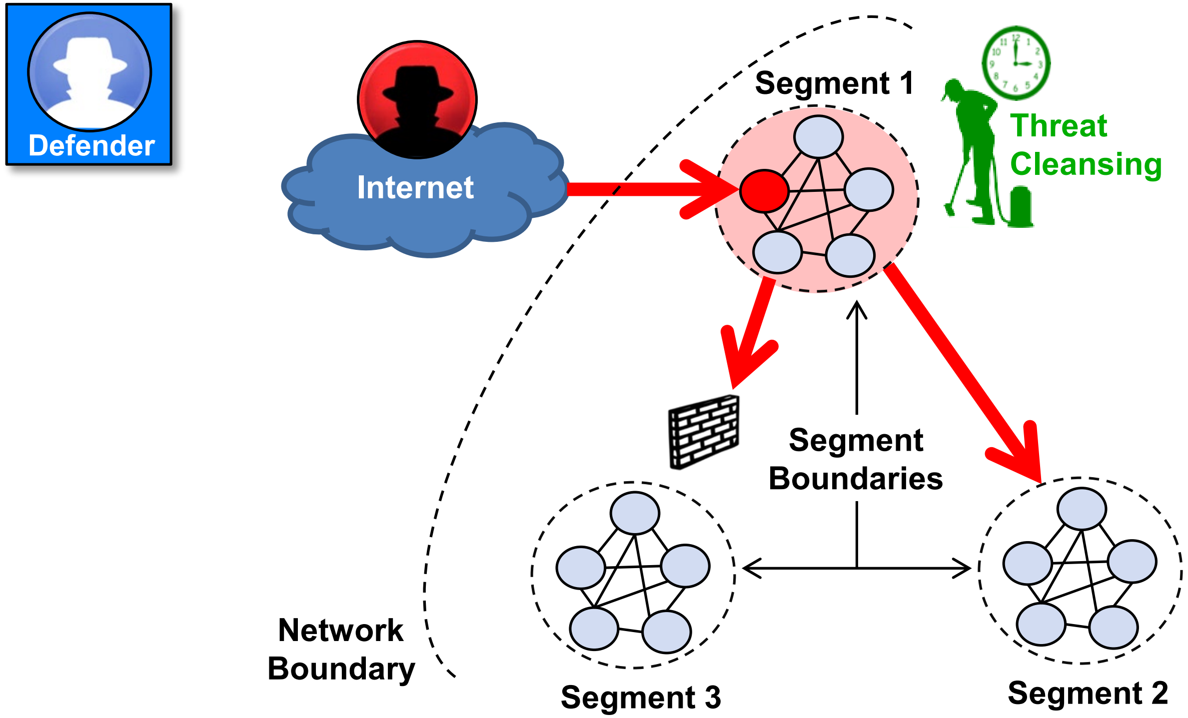




We model this process via Monte Carlo on HPC resources

Network Segmentation





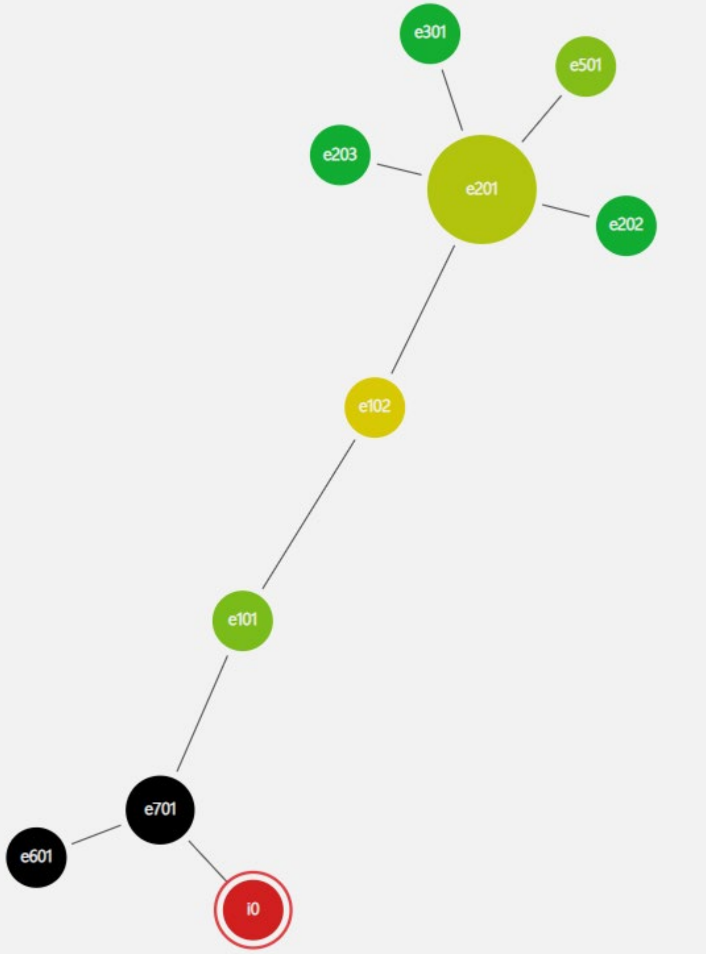
Attacker

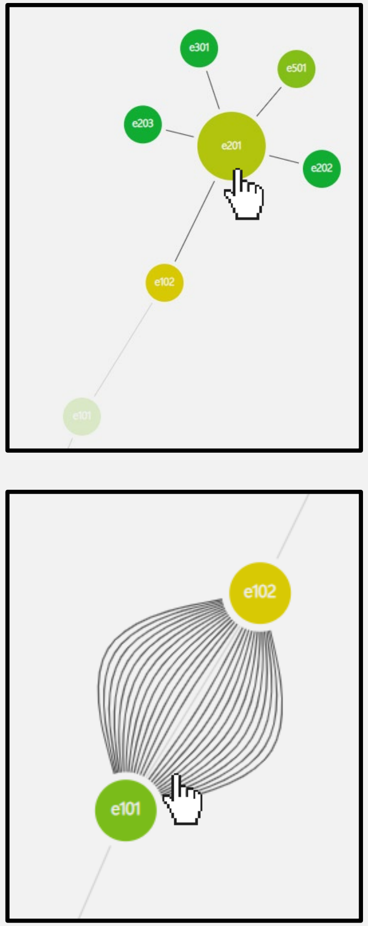
- Exploit vulnerability to penetrate network
- Pivot and spread throughout network

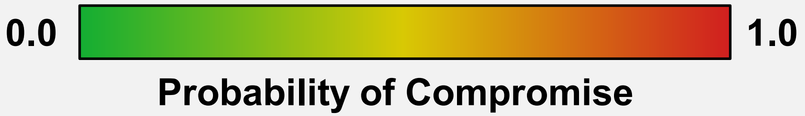
Defender

- Network protected by segmentation architecture
- Communications restricted
- Compromised segments periodically cleansed

Pythia's High Level Network View







Overview

- Initial graph provides a view of all segments of a default topology
- Segments are connected by software services
- Segments are colored by initial risk assessment of contained hosts
- Internet (i0) is fully compromised

Interactions

- Software services visible on hover
- Node connections highlighted on hover

Pythia's Forecast Plot

Overview

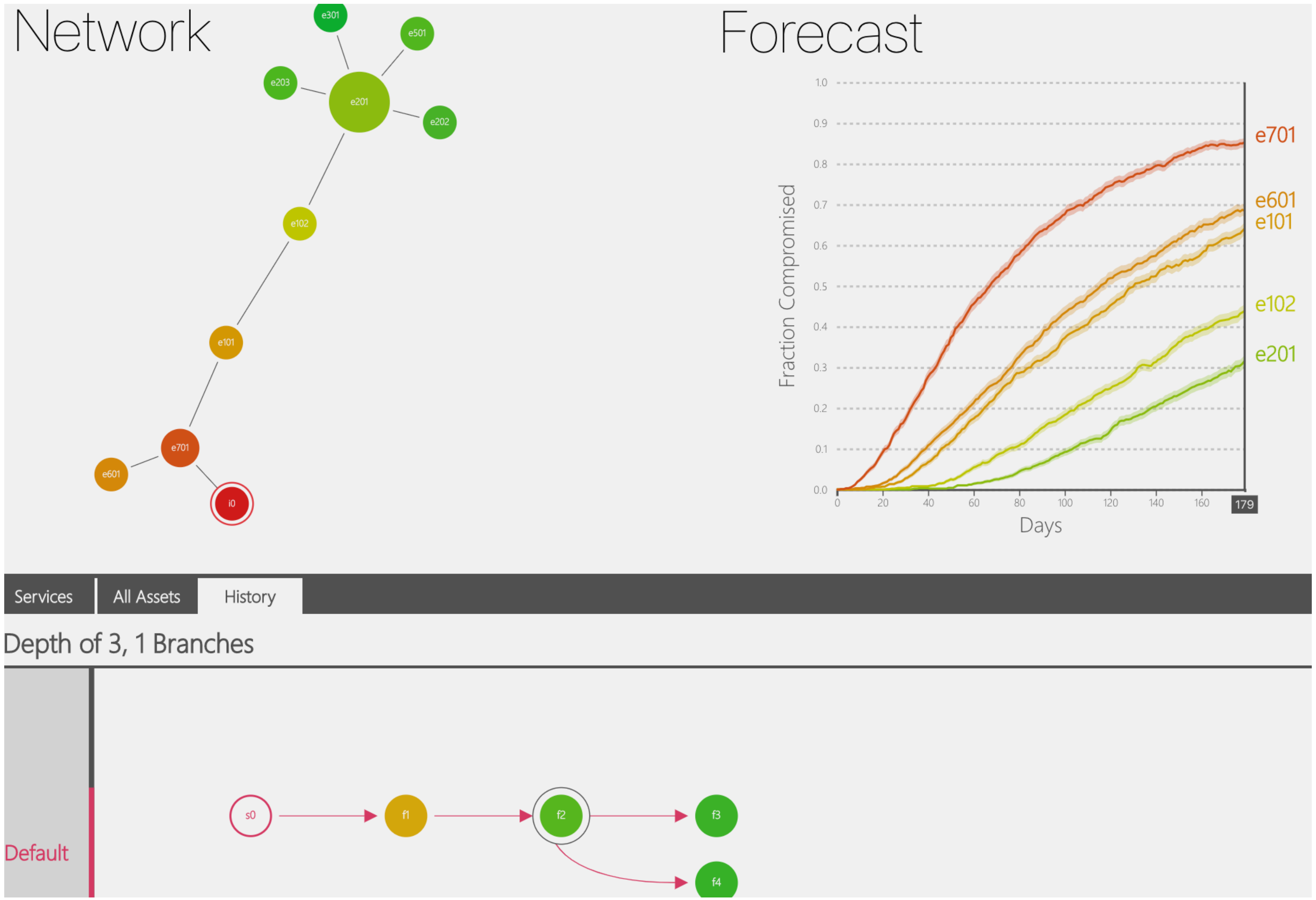
- Initial graph provides a plot of the top 5 most at-risk segments

Interactions

- User scrubbing automatically changes the fill color of the network graph
- This interaction differentiates the initial risk assessment with the forecast
- User enters default patching rates and number of MC trials

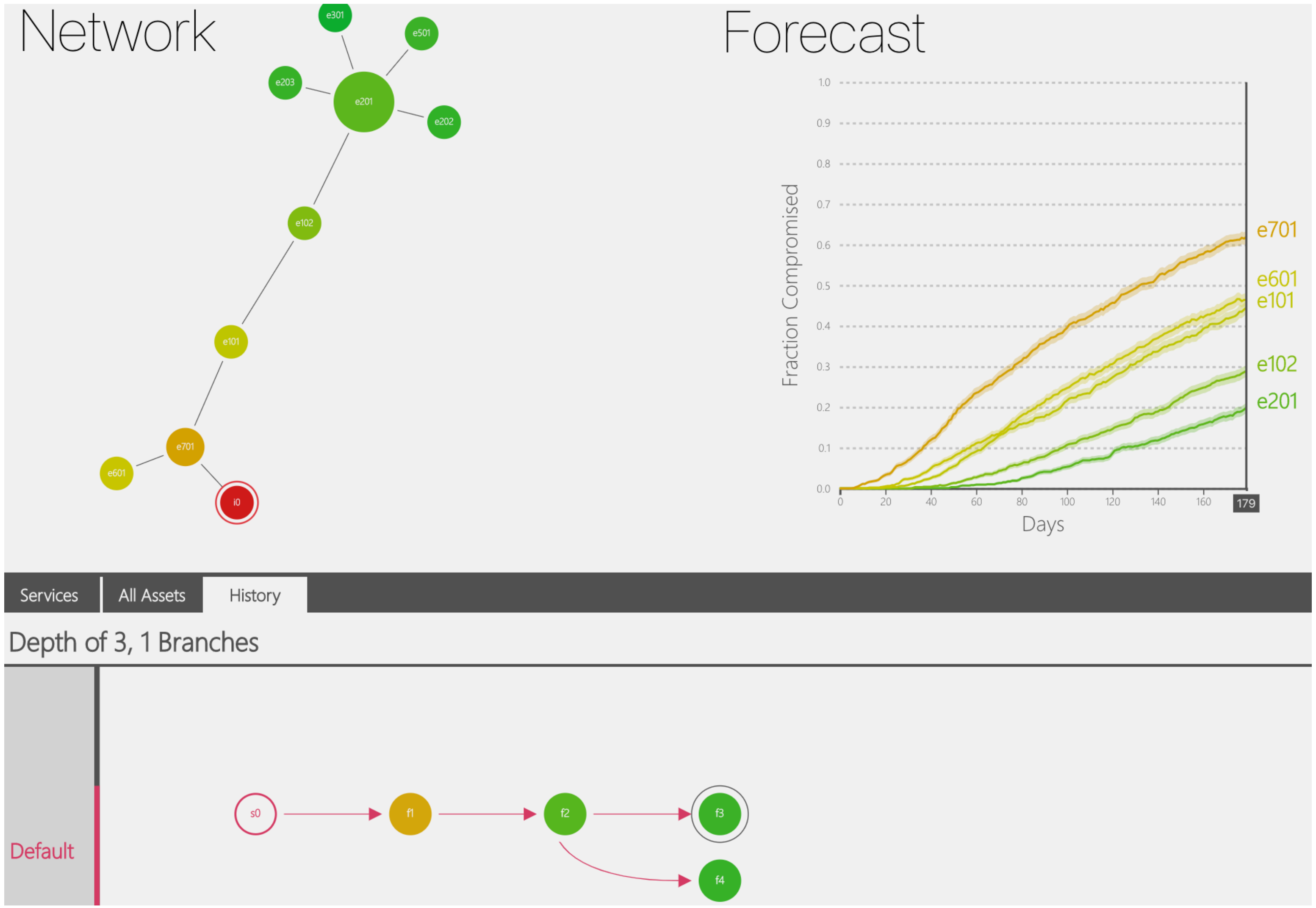
Pythia Forecast and History (1)

- **Example network under nominal conditions**

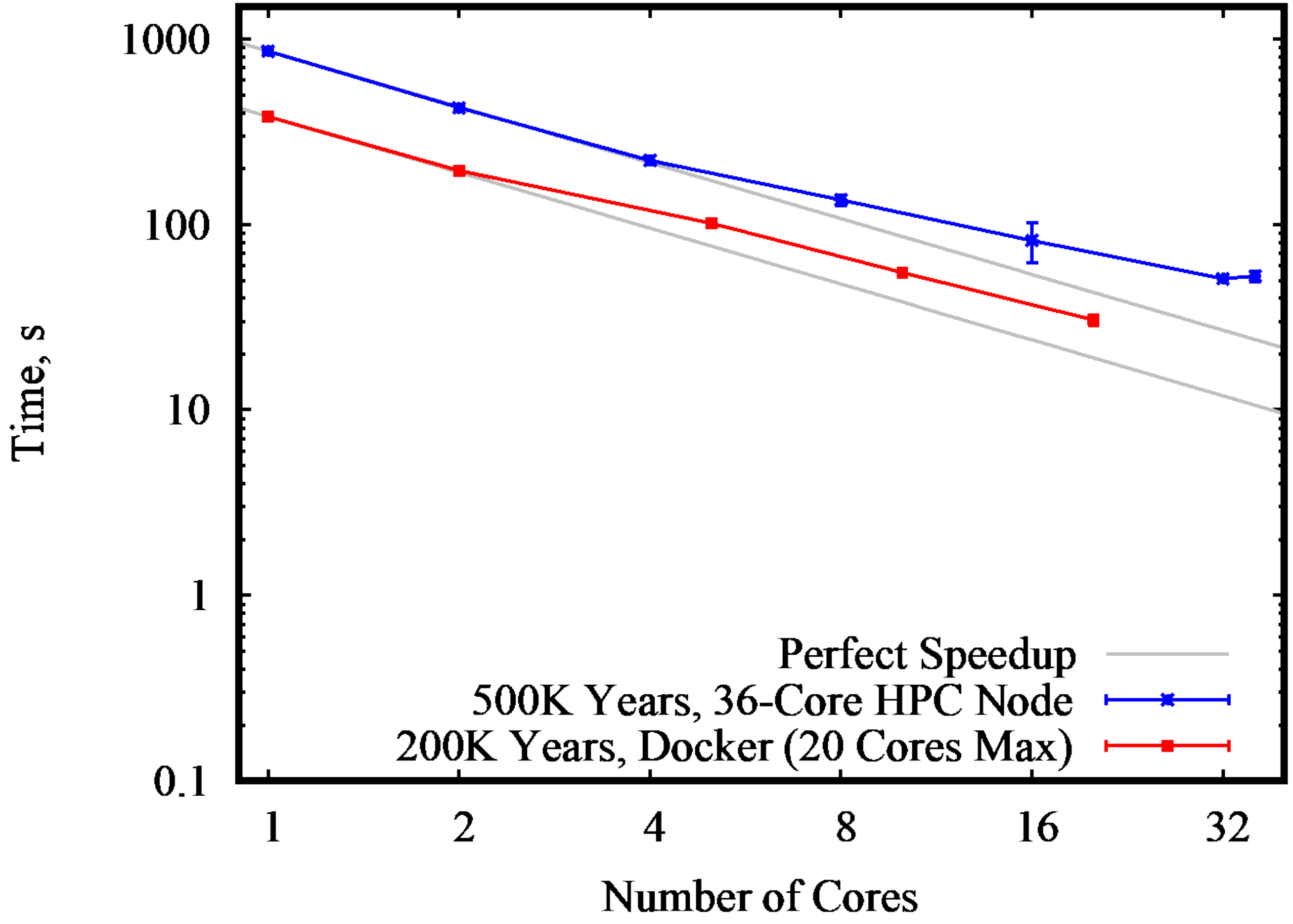


Pythia Forecast and History (2)

- **Same network, but only allowing one browser to connect to i0**
- **This kind of CoA could be implemented by disallowing software or through firewall rules**
- **History tab allows toggling between states**
- **This allows a capability to perform A/B risk assessments**



Simulation Engine Strong Scaling



- **Each network configuration is simulated for one year, 100Ks of times to reduce uncertainty**
- **16.8X speedup on 36 cores (HPC backend) and 12.5X speedup on 20 cores (restricting Docker resources)**
- **Rapid turnaround is essential for operational deployment**

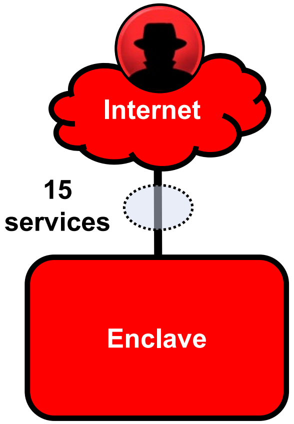
Multicore environments allow simulations to complete in under one minute

Leveraging HPC for CoA Exploration

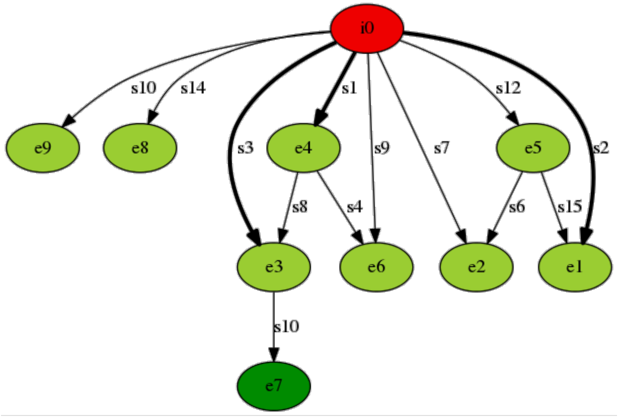
- **Imagine a commander asking a defender for CoAs to mitigate risk given that intelligence indicates an advanced adversary could attack**
- **Manual CoA tradeoffs may be too slow to permit discovery of acceptable risk conditions**
- **As software-defined networking becomes more widespread, it will be possible to *re-segment* an enterprise network such that security risk is reduced at an acceptable level of IT cost.**
- **Pythia accomplishes this via a *genetic algorithm* (GA), implemented in CASCADE**

Example Results Varying Threat Level

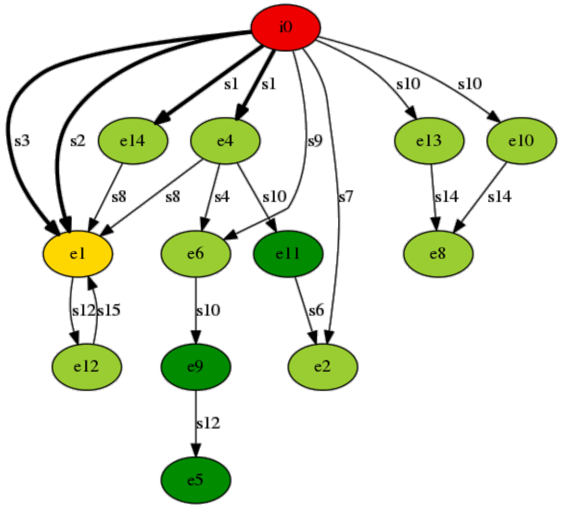
Baseline Architecture



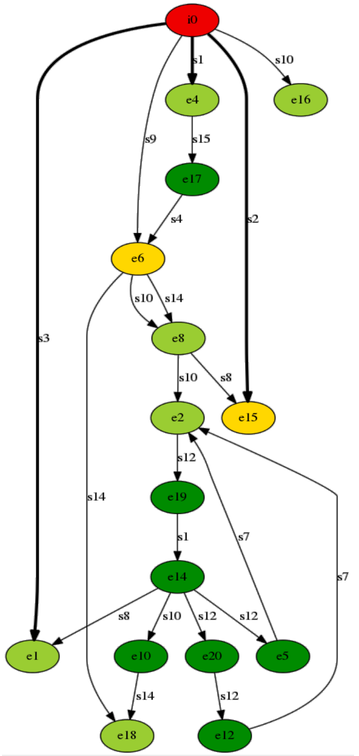
Threat Level 1 Architecture



Threat Level 2 Architecture



Threat Level 3 Architecture

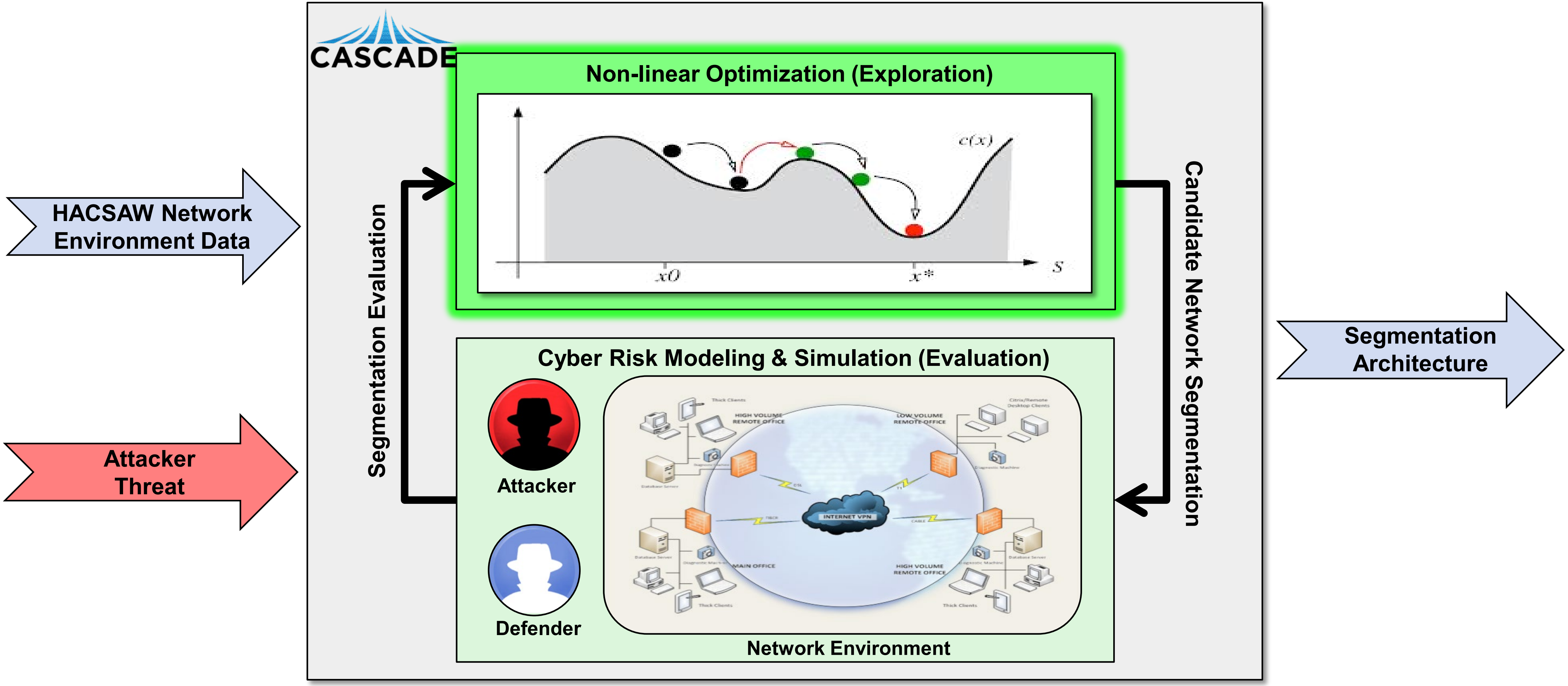


- **Bold** connections denote services that are required to be present in the network.
- Enclaves are colored based on their individual Probability of Compromise (PoC).

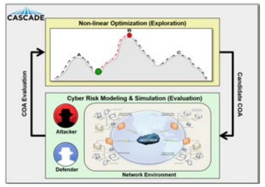
PoC < 0.2 | 0.4 > PoC >= 0.2 | 0.6 > PoC >= 0.4 | 0.8 > PoC >= 0.6 | PoC > 0.8

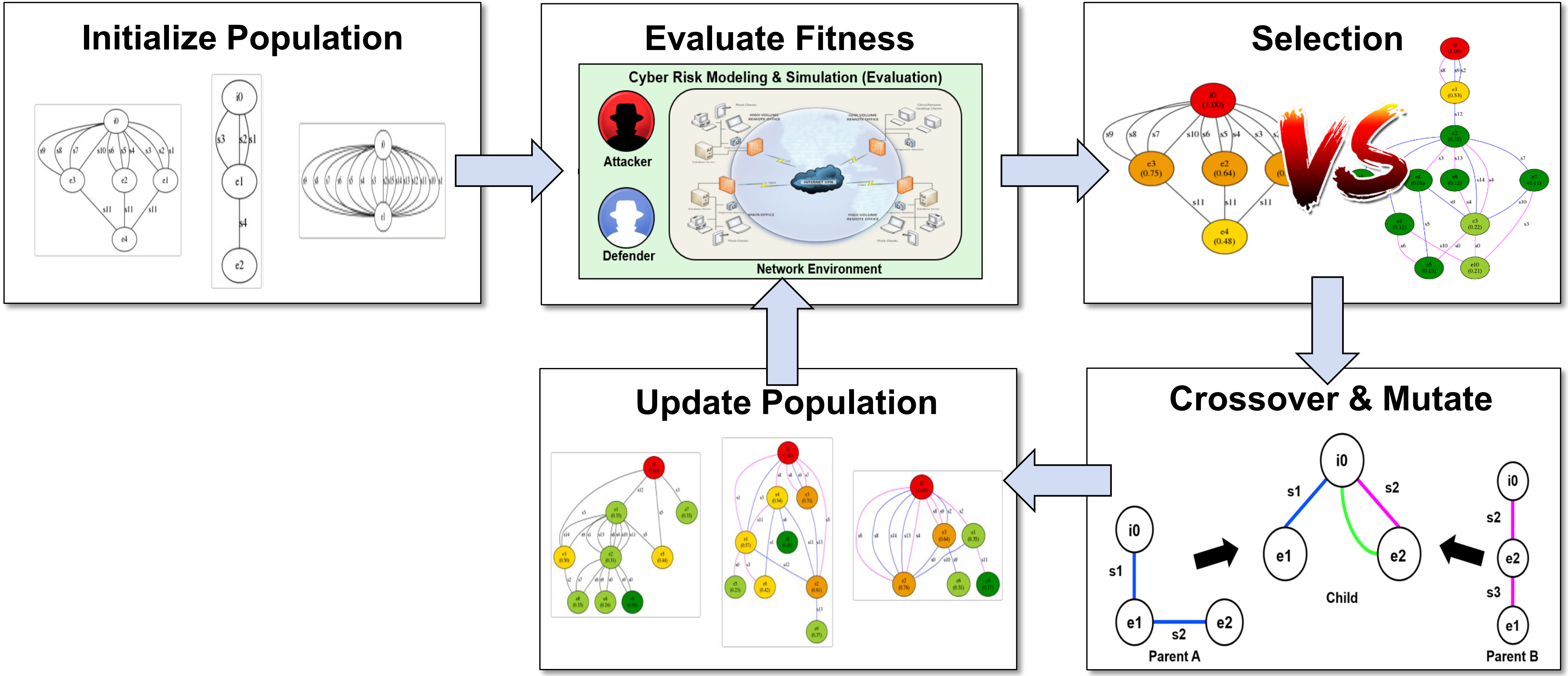
CASCADE can improve baseline architecture to satisfy requirement for acceptable risk and adapt architecture in response to changing threat levels

CASCADE Decision Engine



Genetic Algorithm

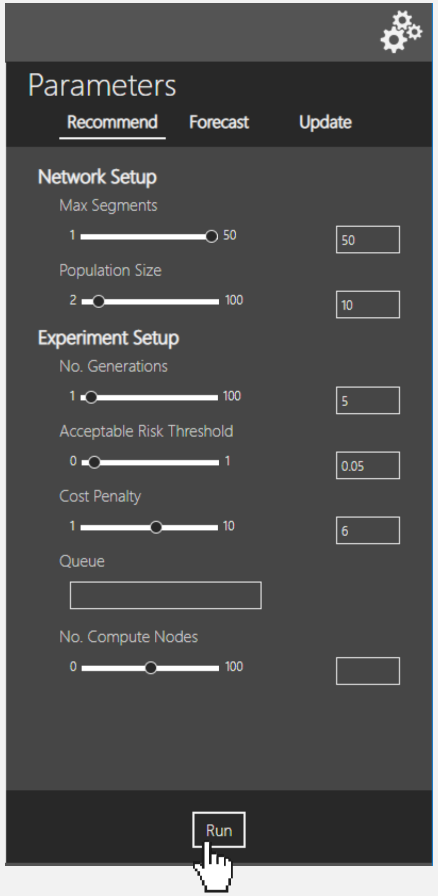


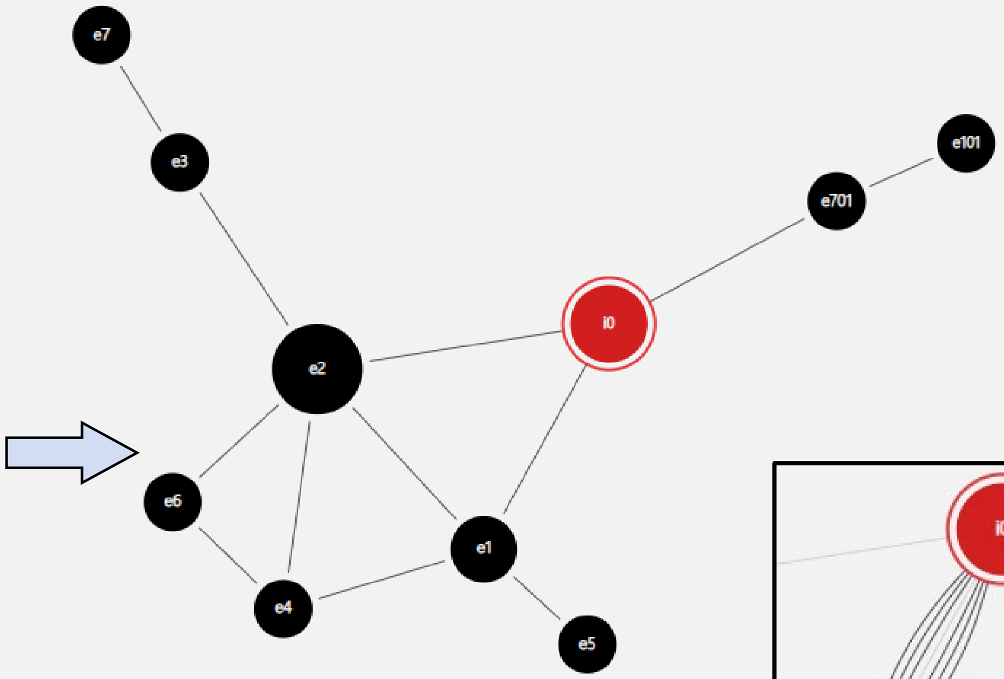


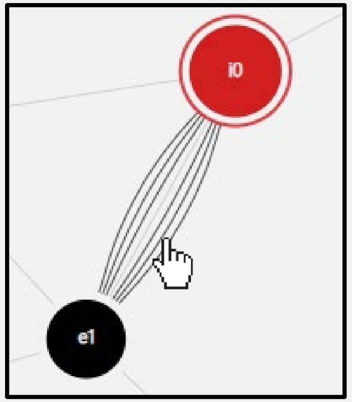
pythia Default "Recommend" GA Parameters

- **Initial population:**
  - 100% Initial Network
- **At each generation:**
  - 80% Tournament Selection
  - 10% Elitism
  - 10% Randomly generated network segmentation
  - Each mutation occurs with .05 probability
- **Fitness Evaluation**
  - 70% Network's Average Probability of Compromise
  - 30% IT Maintenance Cost

Evaluating a New Segmentation

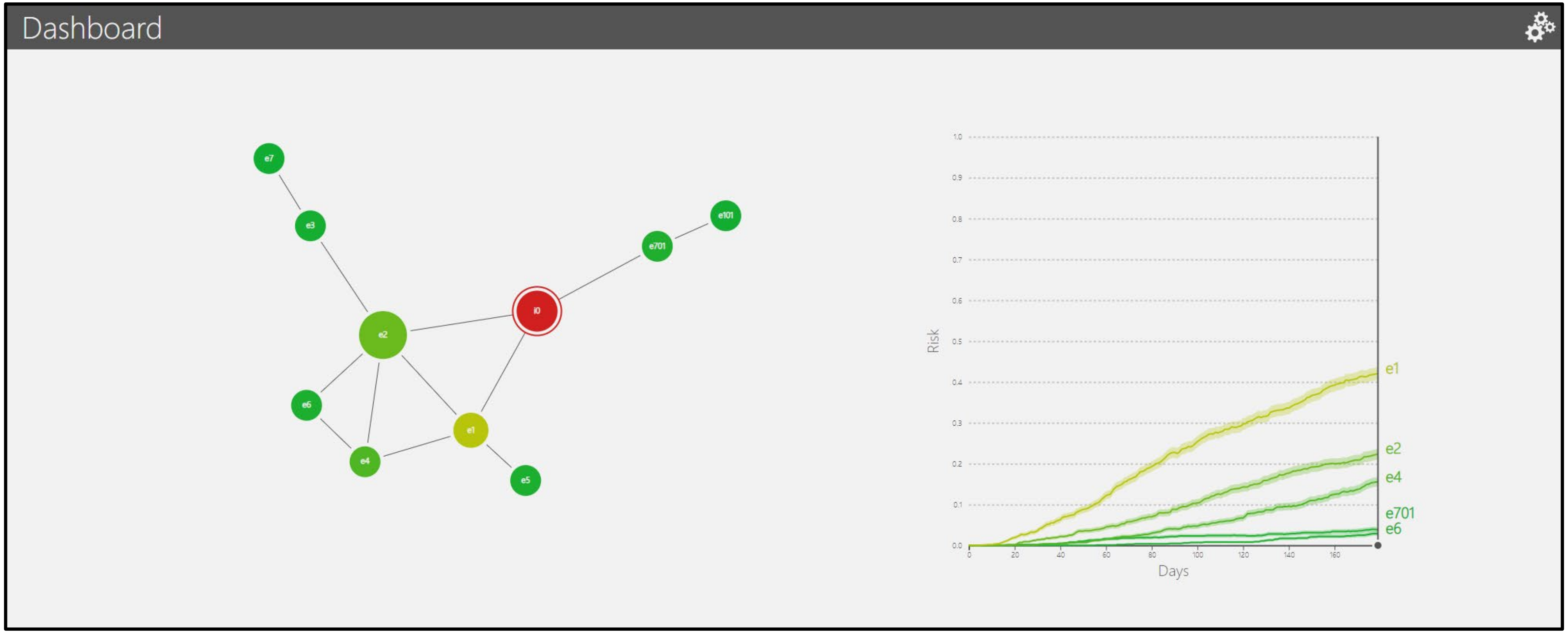






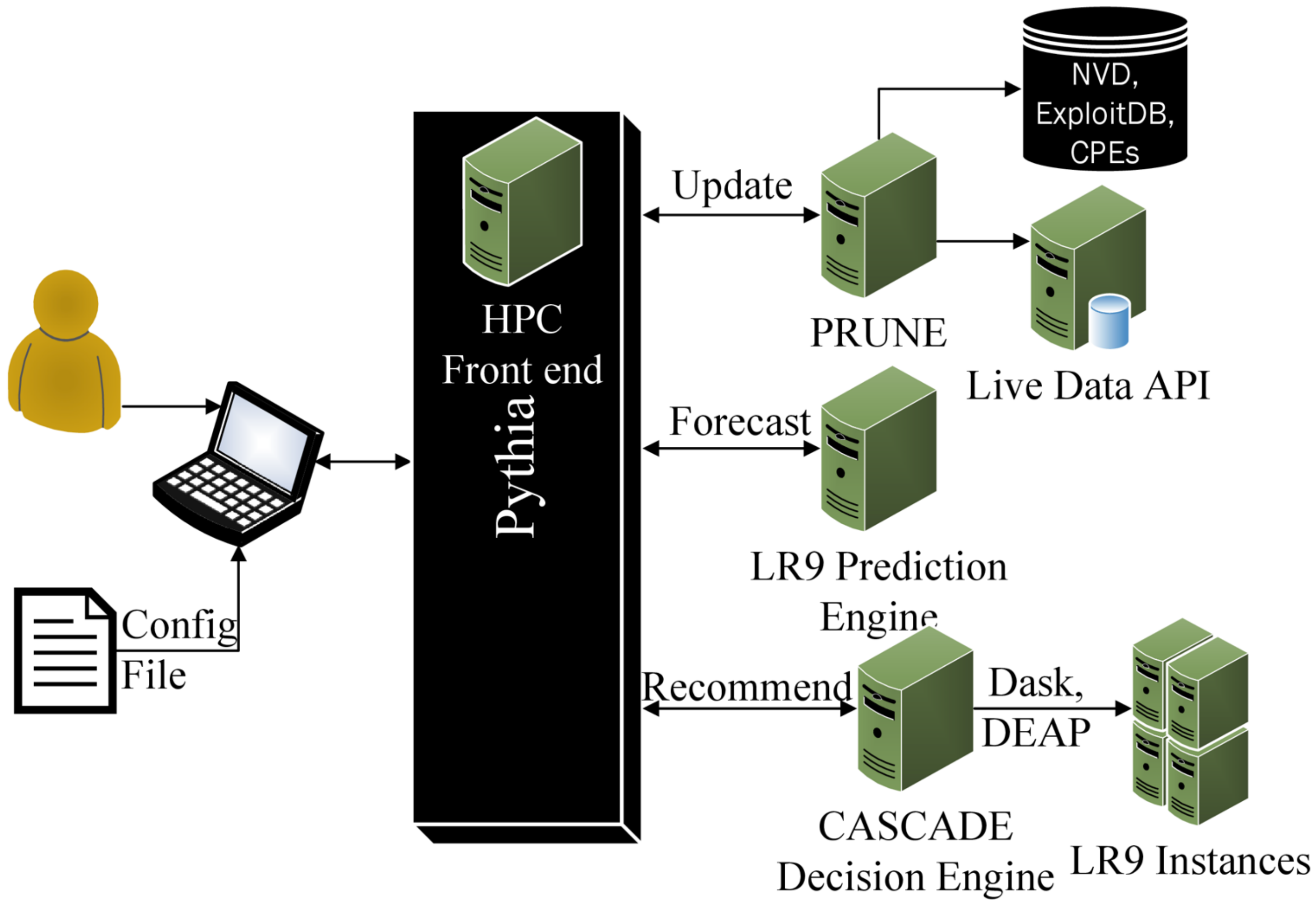
- After a user runs CASCADE from the "Recommend" settings tab, the graph is updated
- With support from high-performance computing (HPC), a recommendation can be generated in minutes

Evaluating a New Segmentation (Cont.)



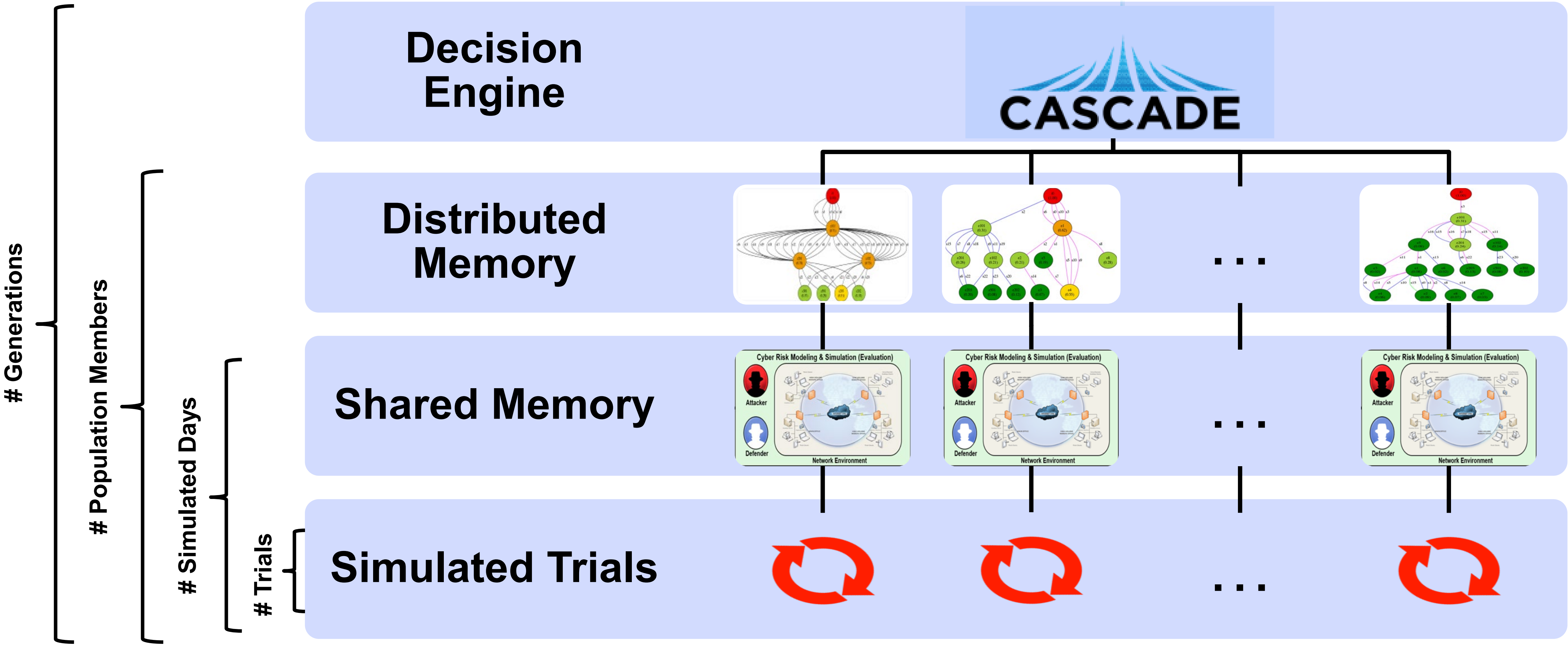
- To evaluate a new CoA, the user may run a new forecast on the network
- The new network yields a much improved risk assessment over the default network

HPC Software Block Diagram

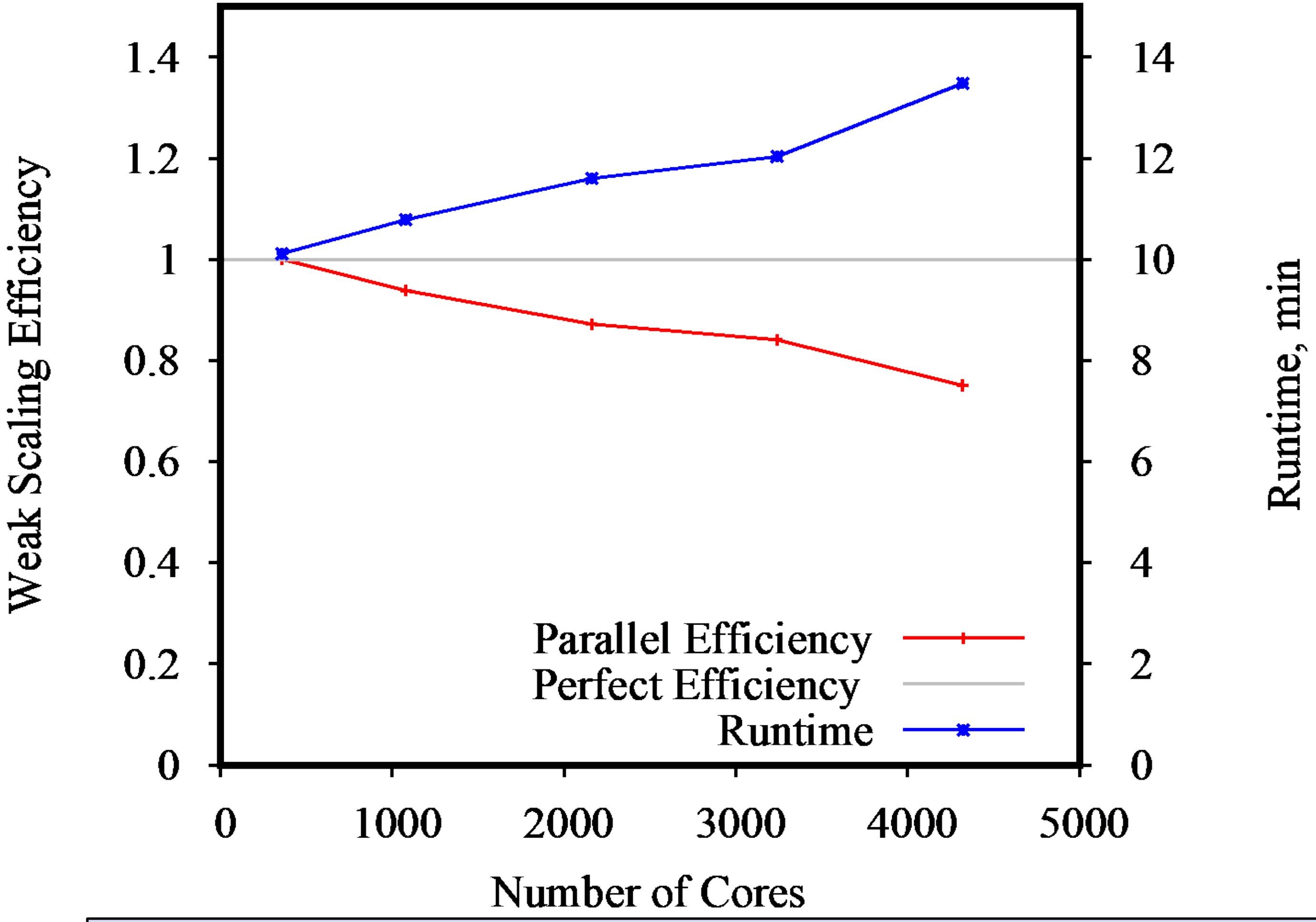


Pythia federates data, prediction, and decision engines to deliver HPC Cyber SA

Pythia HPC Hierarchy



Decision Engine Weak Scaling



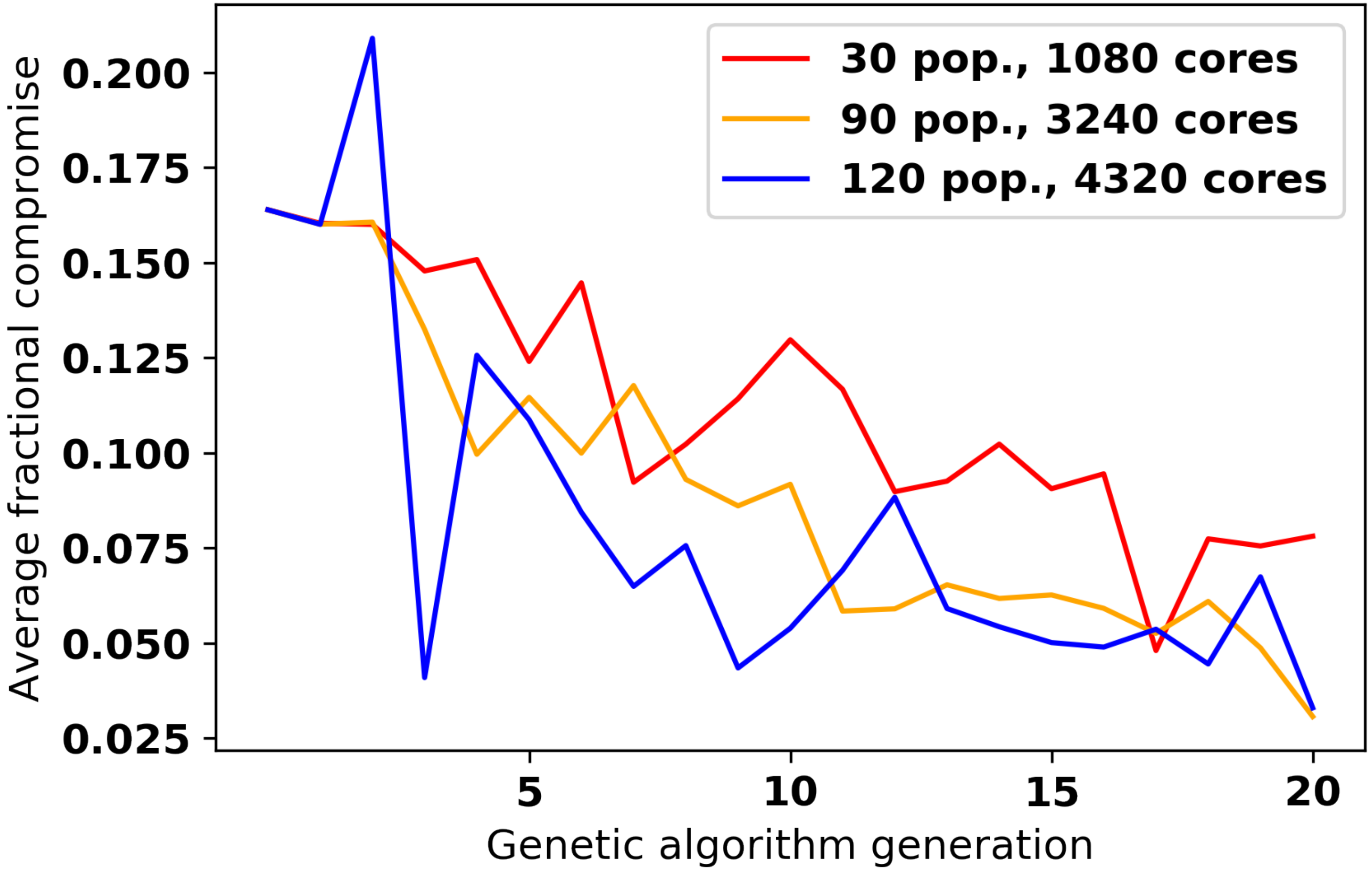
- **Weak scaling: add GA population members and HPC nodes to expand search space**
- **21 Genetic algorithm generations on 120 nodes/population members, each with 100K years, completes in under 15 minutes**
- **Nominal efficiency degradation arising from file I/O between tools**

The decision engine is evaluating over 1 billion simulation-years per hour

Genetic Algorithm Performance

- For the weak scaling study, we fixed the code to use 20 GA generations
- For a threshold of 0.08, GA would have converged earlier (see table)

Pop. size	Iteration #
30	17
90	11
120	3



Adding more population members (HPC nodes) can help GA explore wider and optimize faster

Summary

- **Pythia builds on vulnerability scan data and HPC resources to provide:**
  - quantitative network security measures to warfighters
  - SA of vulnerable software services, hosts, and network segments
  - an ability to enhance cybersecurity in an upcoming timeframe
  - HPC enabled over 1 billion simulation-years per hour for CoA exploration
- **Pythia informs risk projections**
  - Operators can interactively explore the consequences of defensive maneuvers
  - Supports mission planning and analysis of alternatives for network deployment
  - Vulnerabilities can be reprioritized using site-specific threat intelligence

Future Work

- **Model improvements**
  - Enhanced attacker model in pythia (tie exploit development time to risk score)
  - Incorporate other available defensive actions (restricted access)
  - Allow exploitation from within the network
  - Model exploitation at host-level
  - Better account for vulnerability severity levels and effects
  - Enhance configuration data ingestion (firewall, IDS, patch inference)
  - Integrate additional objective functions (mission performance)
  - Incorporate and co-evolve adversarial response
- **Customization by network defenders**
  - Expand course of action recommendation to include additional courses of action
  - Freeze parts of the model where desired, weight mission-relevant services

Acknowledgements

- The Vulnerability Awareness and Recommended Risk Remediation team / co-authors:
  - Jaime Peña, Benjamin Blease, Leslie Shing, Kenneth Alperin, Serge Vilvovsky, Pierre Trepagnier (MIT LL)
  - Neal Wagner (Now at Systems and Technology Research)
  - Leslie Leonard (U.S. Army Engineer Research and Development Center)
- High Performance Computing Modernization Program
  - For supercomputer time, HPC assistance, supporting the HACSAW data API, and supporting this effort demonstrating the use of HPC for Cyber SA
  - Special thanks to William Glodek and Ben Parsons

Questions?

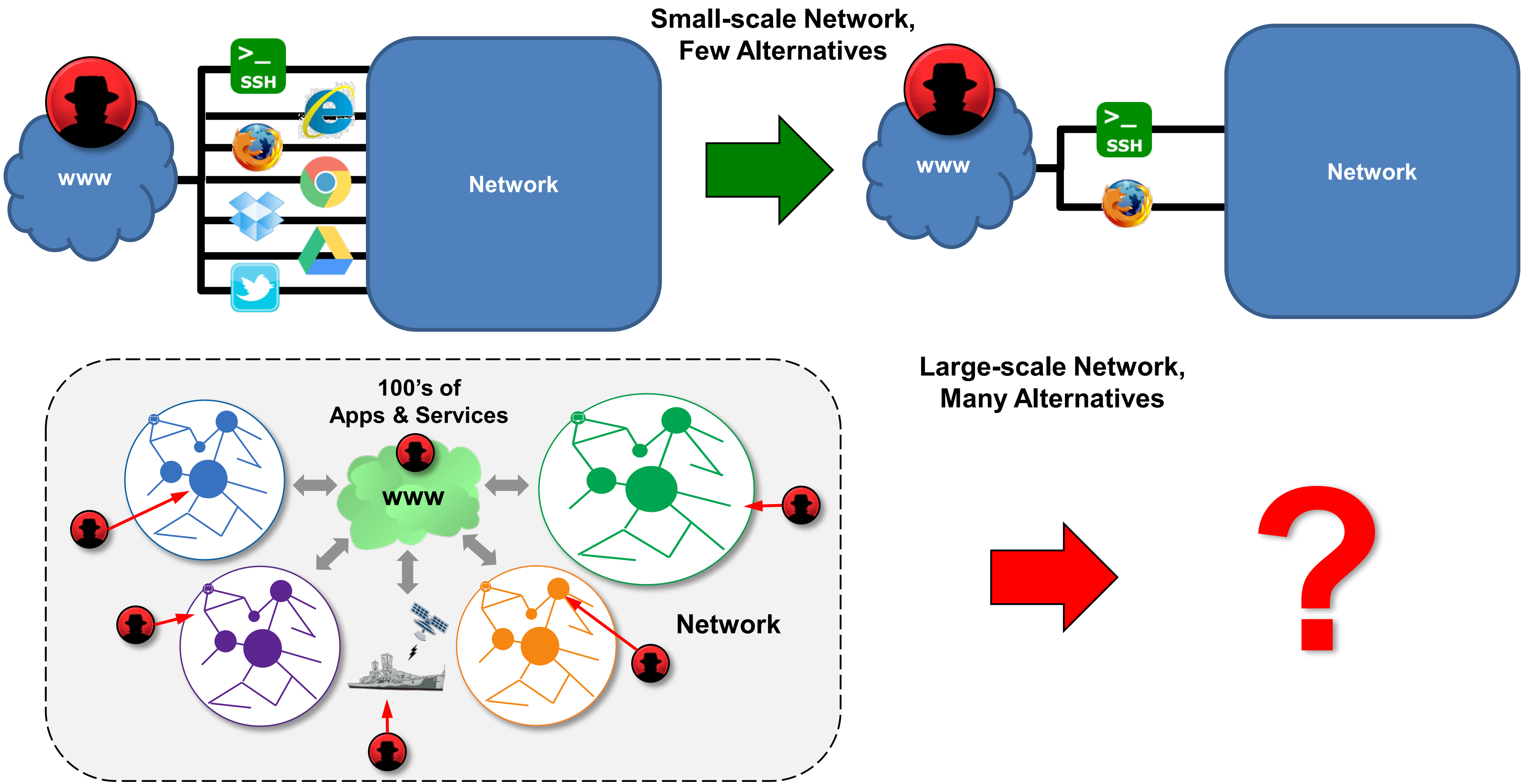
Allan Wollaber
Allan.Wollaber@ll.mit.edu

Outline

- Background
- Methods
- Results
- Summary
- Backup

Backup

Network Segmentation Is Difficult for All But Small-scale Networks



Measuring Security and Cost

- **Security is measured as the expected probability of enclave penetration by attacker**
  - Values in [0,1]
  - Lower values mean lower security risk
- **Cost is characterized as IT maintenance effort**
  - More enclaves = more cost to maintain
- **Exponential function utilized to capture cost increase as the total number of enclaves increase**
  - Normalized to [0,1]
  - Lower values mean lower cost
- **Combined risk is computed as a weighted average of security and cost**

Security Measure:
Expected Probability of Enclave Penetration

$$Sec(env,s) = \frac{1}{|encls(s)|} \sum_{e \in encls(s)} P_{penetrate}(e)$$

Cost Measure:
IT Maintenance Effort of *N* Enclaves

$$C(env,s) = \frac{e^{N+k/M} - 1}{e^{k} - 1}$$

Combined Risk Measure

$$R(env,s) = w_1 \cdot \mathrm{Sec}(env,s) + w_2 \cdot C(env,s)$$

pythia


pythia v1.1.4
Dark Mode
Dashboard
A
B
C
e301
e501
e203
e201
e202
e102
e101
e701
e601
i0
Risk
Days
e701
e601
e101
e102
e201
Services
All Assets
History
Default Topology, 56 Services
Name | Vulnerability Interval | Exploit Delivery Interval | Exploit Development Interval | Patch Interval | Delete
citrix_receiver | 140 | 76 | 0 | 25 | Delete
java | 24 | 76 | 0 | 25 | Delete
notepad++ | 1394 | 76 | 0 | 25 | Delete
skype_business | 205 | 76 | 0 | 25 | Delete
skype_business_plug-in | 205 | 76 | 0 | 25 | Delete
spark | 84 | 76 | 0 | 25 | Delete
rhel_cups | 512 | 76 | 0 | 25 | Delete
rhel_mongodb | 152 | 76 | 0 | 25 | Delete
rhel_postfix | 441 | 76 | 0 | 25 | Delete

D


Services
All Assets
History
190 Assets
Segment | Name | IP | Score | Vulnerabilities | Exploit Available
e101 | HOST_NAME | 127.0.0.1 | 0 | 0 | No
e101 | HOST_NAME | 127.0.0.1 | 0.47 | 6 | Yes
e101 | HOST_NAME | 127.0.0.1 | 0.47 | 6 | Yes
e101 | HOST_NAME | 127.0.0.1 | 0.19 | 2 | Yes
e101 | HOST_NAME | 127.0.0.1 | 0.19 | 2 | Yes
e102 | HOST_NAME | 127.0.0.1 | 0.94 | 26 | Yes
e102 | HOST_NAME | 127.0.0.1 | 0.47 | 6 | Yes
e102 | HOST_NAME | 127.0.0.1 | 0.35 | 4 | Yes
e102 | HOST_NAME | 127.0.0.1 | 0.77 | 14 | Yes

E


Services
All Assets
History
Depth of 2, 2 Branches
Default
Topo0
s0
f1
r1
r2