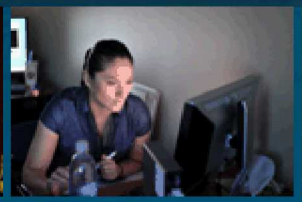
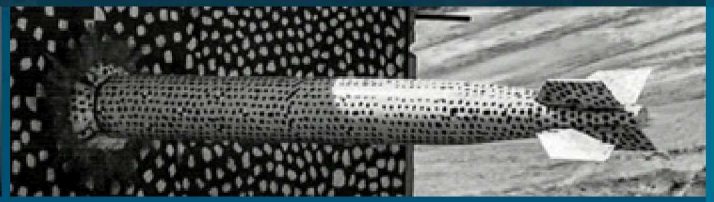
SAND2018-11779C

Performance-Based Cyber Resilience Metrics: An Applied Demonstration Toward Moving Target Defense





PRESENTED BY

Christine Lai

Shamina Hossain-McKenzie Eric Vugrin Adrian Chavez

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

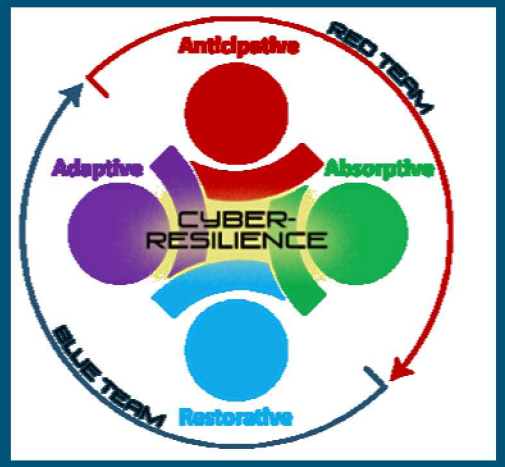
ADDSec

Artificial Diversity and Defense Security

Grid WANs have predictable communication paths and static configurations

To introduce unpredictability and enhance situational awareness, Chavez et al. developed the ADDSec tool which **leverages moving target defense (MTD)**

- **Anticipates and adapts** against reconnaissance and Ethernet-based attacks using software-defined networking (SDN)
- Enables automatic reconfiguration of the system through **IP randomization,** port hopping, and instruction set randomization
- Detects attacks using machine learning and notifies SDN controller to randomize



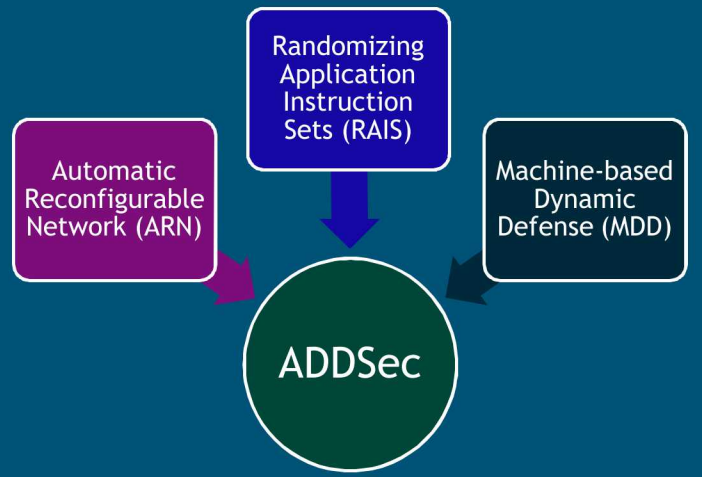


Image source: Chavez et al., 2017

Resilience Metrics

Measurement of resilience costs considers:

- Systemic Impact (SI): cumulative impact that a disruption has on system performance

$$SI = \Sigma_{i=1}^{N}[TSP(t_i) - SP(t_i)](t_i - t_{i-1})$$

- Total Recovery Effort (TRE): total resources used for recovery efforts post-disruption

$$TRE = \Sigma_{l=1}^{M}[RE(t_l)](t_l - t_{l-1})$$

Calculate the recovery-dependent resilience (RDR) costs:

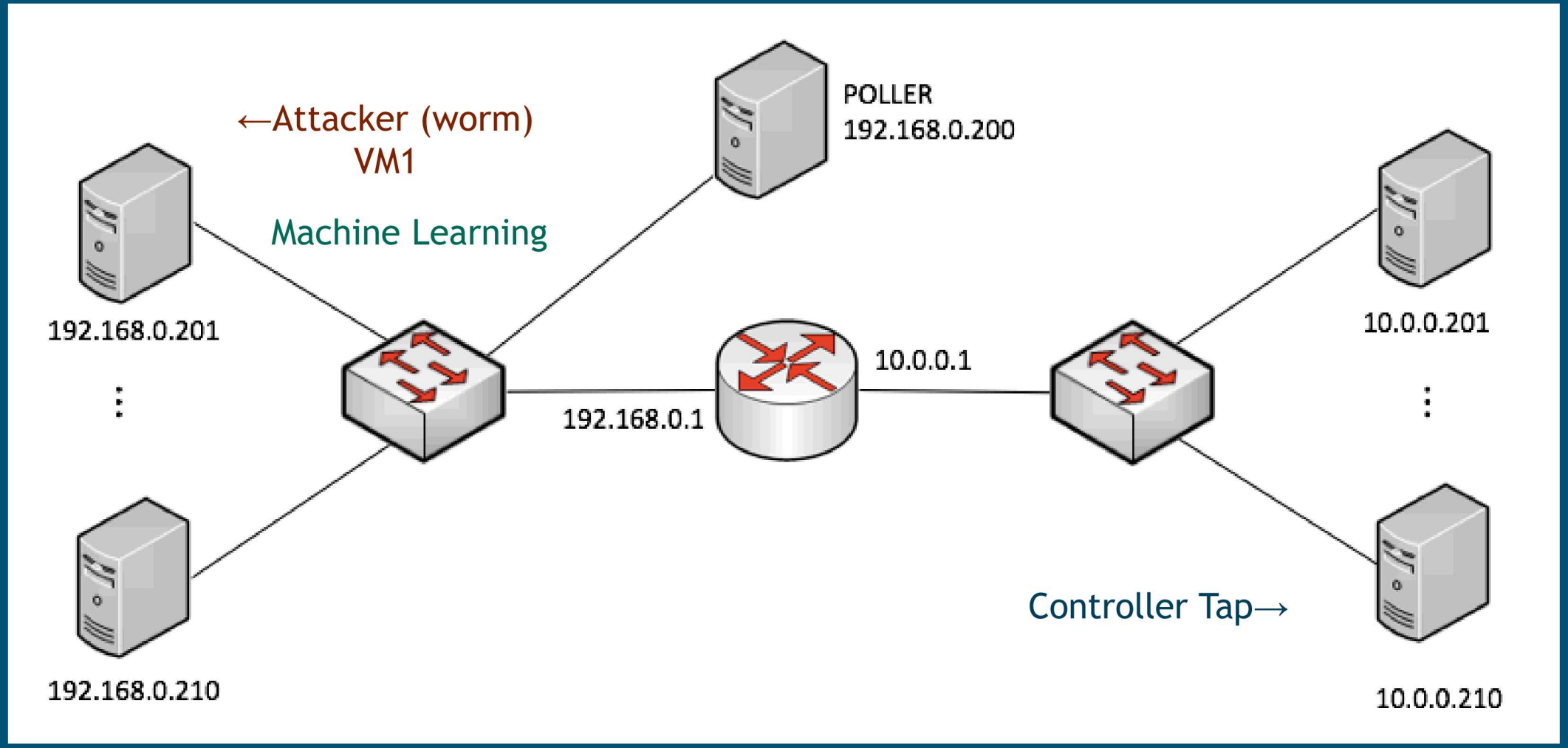
- Takes into account the effect the different recovery activities have

$$RDR = \frac{SI + \alpha \cdot TRE}{Norm}$$

Research Questions

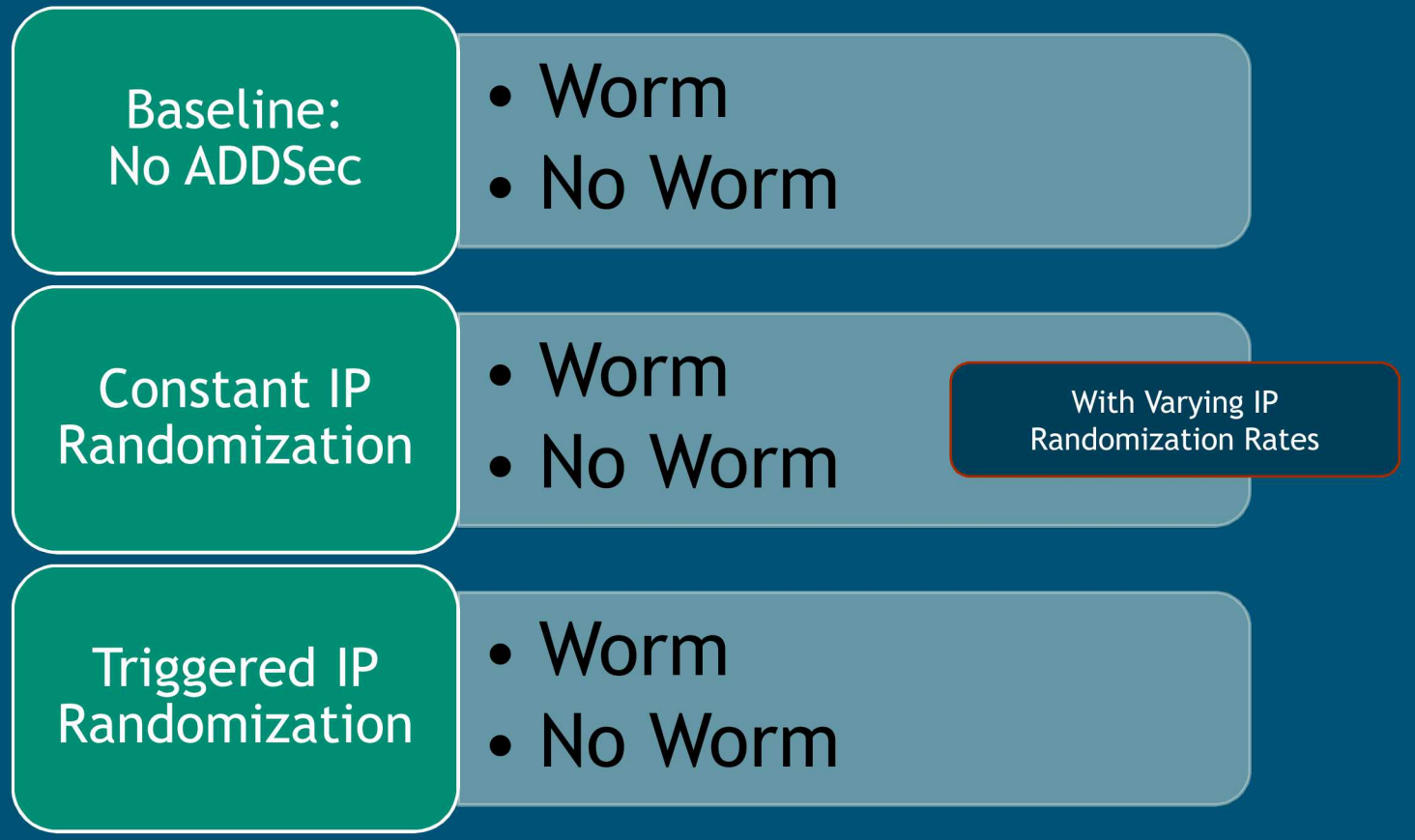
Key Questions:	1.1. Does ADDSec increase resilience of the system during an attack, specifically during reconnaissance?
	1.2. What performance does the system exhibit under different IP randomization rates?
	1.3. What performance does the system exhibit under different IP randomization rates during an attack?
	1.4. Are machine learning triggers effective for this type of attack?
	1.5. Do our resilience metrics provide useful insight into the effectiveness of ADDSec?

Experimental Setup
Demo-Case



Experiment Plan

ADDSec Modes and Attack Presence



- Disturbance: Worm deployed on (an initially single) host(s) attempting to ping addresses and make connections
- Ran 10 trials for each case

Performance Metrics of Interest
Computing SI and TRE

Systemic Impact (SI)

- *System Metrics:*
  1. Hosts Not Infected (#)

Total Recovery Efforts (TRE)*:

- *System Metrics:*
  1. Latency (s)
  2. Retransmissions (#)
  3. Dropped Packets (#)

*Latency weighted most heavily, then dropped packets, and then retransmissions

Image source: ITPro, 2010 and Contemporary Research, 2017

Results Summary
System Metrics

		Frequency of IP Randomization									
Average over 10 trials (1000s/trial)		None	ML	1s	4s	8s	16s	32s	64s	128s	256s
# Host Infections	*No Worm*	-	-	-	-	-	-	-	-	-	-
	Worm	20	3	2.8	3.4	4.8	4.9	4.9	7.9	8.9	9.8
Latency	*No Worm*	29.93	37.2	349.34	394.71	699.11	591.89	TBD	422.1097	48.88403	420.31
	Worm	729.91	698.92	346.22	733.84	1000.42	1148.1	997	1187.3	1559.14	2351.07
Retransmissions	*No Worm*	6039	5928.7	37.2	37.2	37.2	37.2	TBD	4291.8	6887	2681.3
	Worm	5417	2267.8	1966.1	2151.1	2451.9	2839.5	3911.3	6297.6	7182.3	3911.3
Dropped Packets	*No Worm*	0	0	0.1	0	0.1	0	TBD	0	0	0
	Worm	0	0.3	1	0.7	0.6	0.1	0	0.1	0	0

Results Summary

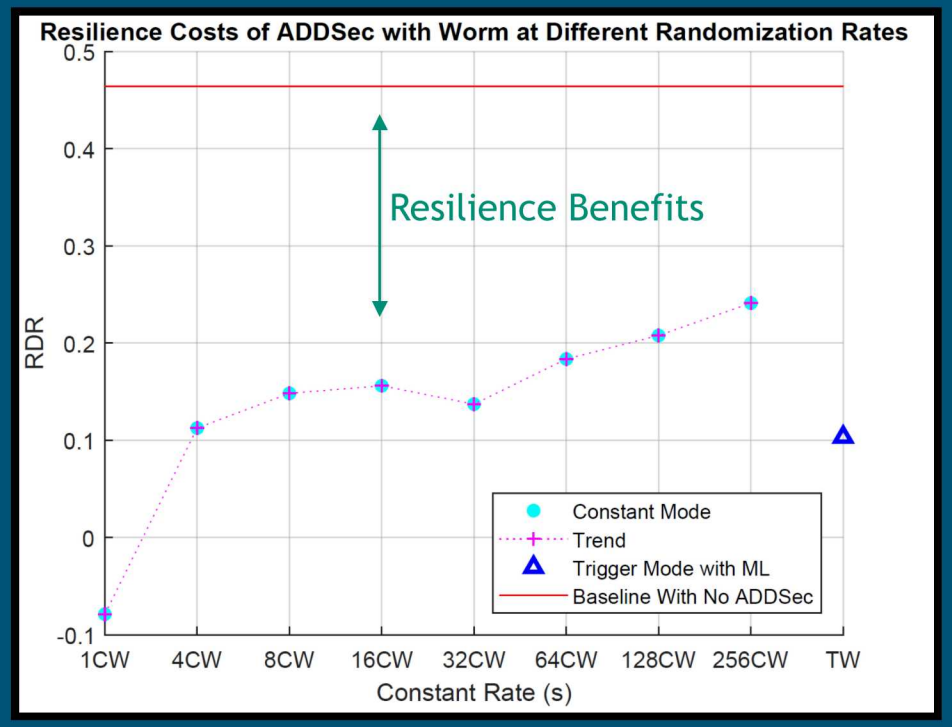
Resilience Metrics

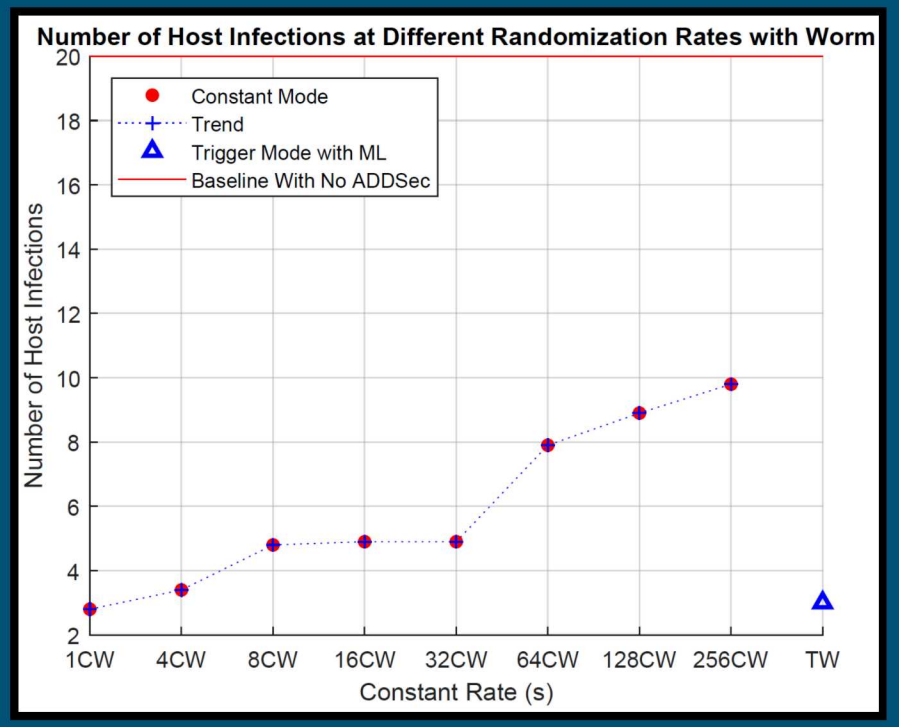
	Frequency of IP Randomization									
Average over 10 trials (1000s/trial)		None	ML	1s	4s	8s	16s	32s	64s	128s	256s
SI	*No Worm*	0	0	0	0	0	0	0	0	0	0
	Worm	0.65146	0.05773	0.05378	0.06091	0.08202	0.08524	0.08373	0.1331	0.15133	0.16696
TRE	*No Worm*	-0.00042	-0.00235	-0.00341	0.01331	0.02631	0.01751	TBD	0.0202	0.00094	0.0442
	Worm	-0.1872	0.04558	0.02497	0.05158	0.06614	0.07078	0.05336	0.0504	0.05643	0.07413
RDR	*No Worm*	0.00042	-0.00235	-0.00341	0.01331	0.02631	0.01751	TBD	0.0202	0.00094	0.0442
	Worm	0.46426	0.1033	-0.07874	0.11247	0.14817	0.15602	0.13709	0.18352	0.20777	0.24108

Results

Key Question: 1.1 Does ADDSec increase resilience of the system during an attack, specifically during reconnaissance?

Yes! ADDSec improves resilience significantly.

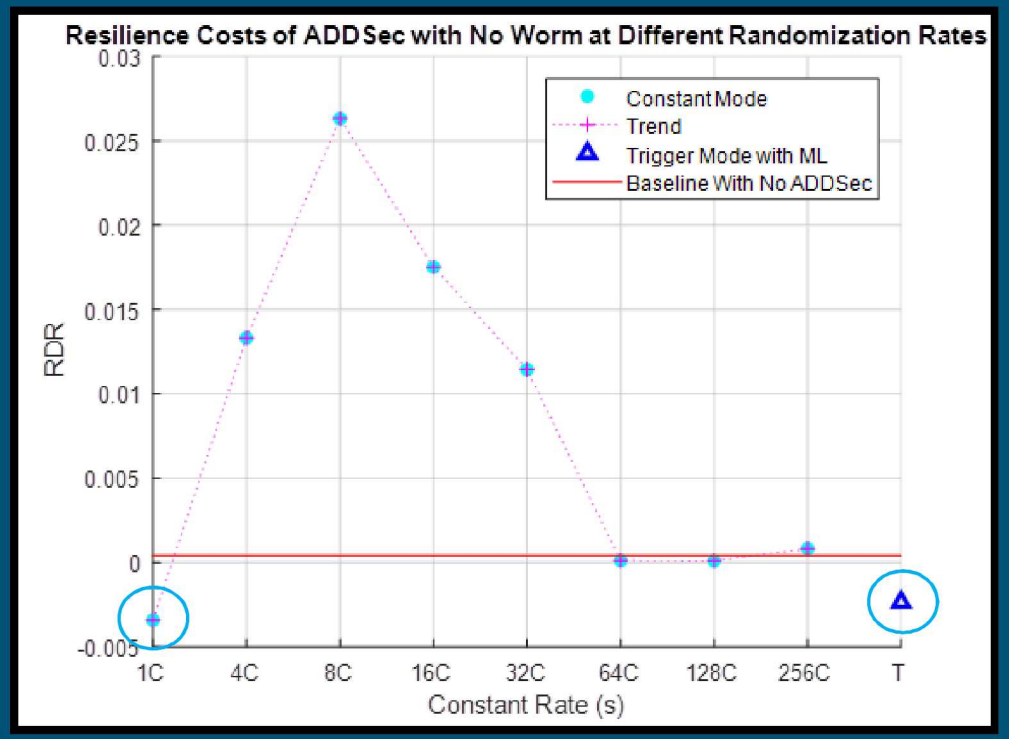




Results

Key Question: 2 What performance does the system exhibit under different IP randomization rates?

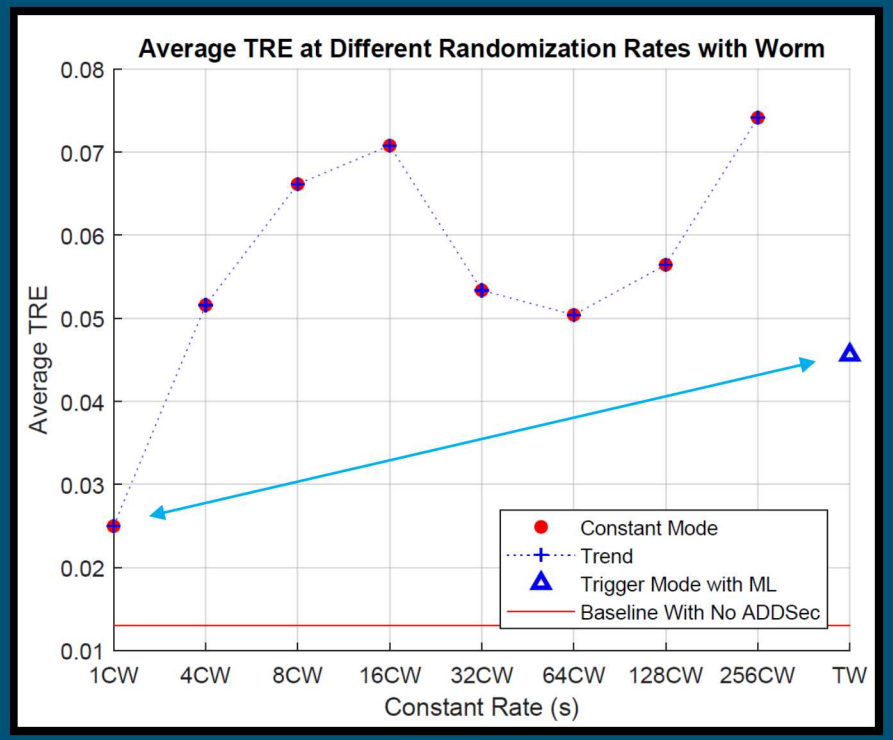
Constant 1s and Trigger Mode lower performance losses.



Results

Key Question: 1.3 What performance does the system exhibit under different IP randomization rates during an attack?

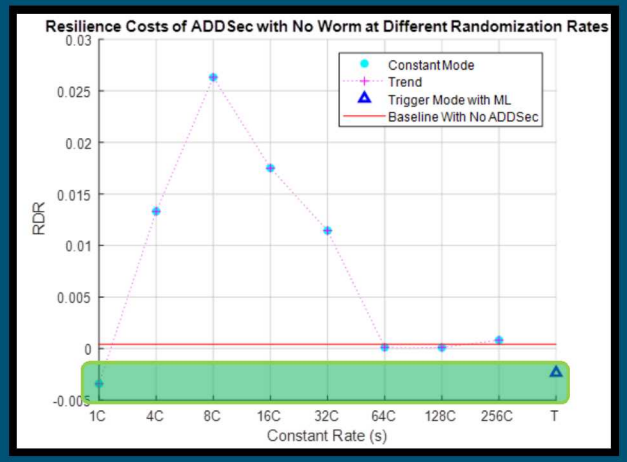
Constant 1s and Trigger Mode low performance overhead.

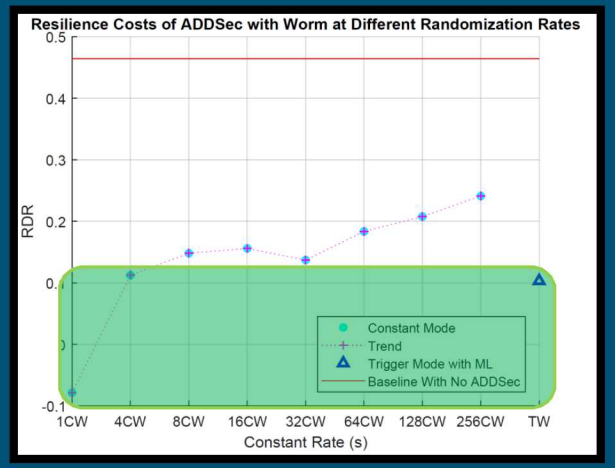


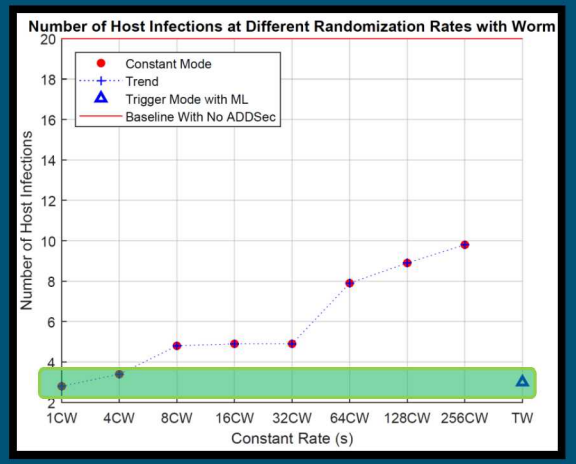
Results

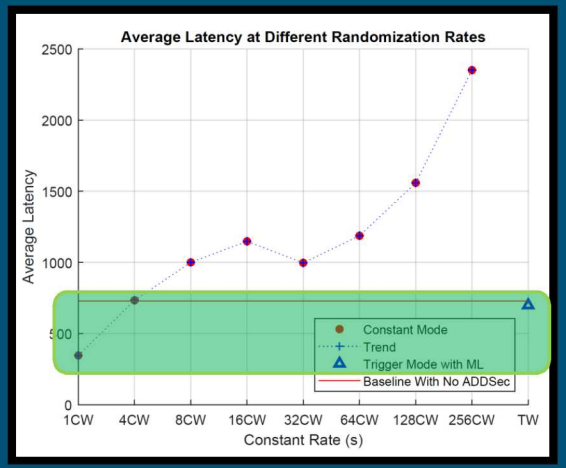
Key Question: 4 Are machine learning triggers effective for this type of attack?

Triggered randomization exhibited similar behavior to faster randomization rates; Constant 1s Mode always outperforms.





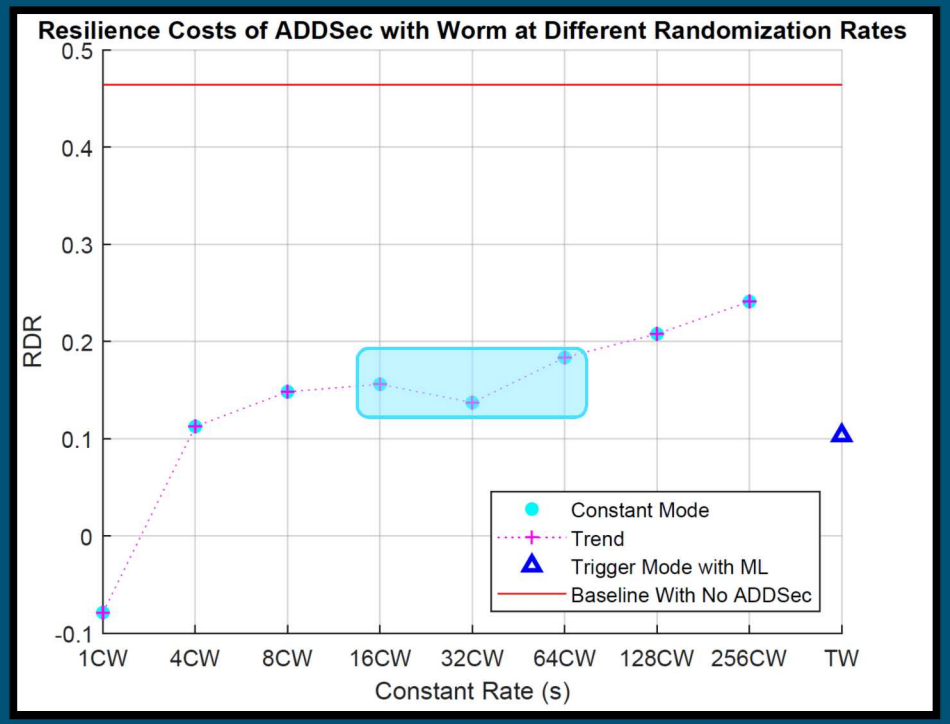




Results

Key Question: 5 Do our resilience metrics provide useful insight into the effectiveness of ADDSec?

Trends are seen in relation to ADDSec randomization rate/strategy; found that Constant 1s Mode most effective.



Key Takeaways

Resilience analysis provides useful insight into ADDSec performance and optimal modes

- SI metric captures infection impact to system dynamically, over time
- TRE metric can be tuned to give more weight to important quantities (e.g., latency > retransmits)
- RDR provides more granular insight that might be missed with only intuition (e.g., 32s case)

Automated triggers can be effective

- Reconnaissance activity is stopped even during period of the randomization rate
- Higher resilience than constant rate
- Caveat: algorithms need to be tuned to detect the attack

IP randomization is effective but subject to variability

- Quantitative analysis shows that faster randomization rates improve resilience on average
  - Increasing randomization decreases number of infected hosts and time to first infection
- Stochastic behavior means that there is no guarantee of improved resilience with faster randomization

Thanks! Questions?

Summary

Energy systems target of cyber attacks; WANs predictable and static

Does moving target defense effectively defend against reconnaissance and Ethernet-based attacks?

ADDSec: Artificial Diversity and Defense Security (Chavez et al., 2016) employs MTD

- Automatically reconfigures system with IP randomization and port hopping
- Can detect attack and then randomize using machine learning algorithms

Does ADDSec make the system more resilient?

Using quantitative resilience metrics and analysis, results indicate:

ADDSec does improve system resilience during a reconnaissance attack!

ADDSec is worth the cost of implementation for our target system.

Cyber Resilience

Many critical systems are the target of evolving, sophisticated attacks

- Cannot stop every attack – need to improve **cyber resilience**

Vugrin et al. on resilience:

- Given one or more disruptive event(s), resilience describes the system's ability to reduce the magnitude and duration of deviation from targeted performance levels

Quantitatively evaluate resilience features such as ADDSec to make informed decisions by examining:

- Effectiveness of tool during a disruption
- Impact on normal system operations
- Resilience costs of different implementation strategies

Informally, cyber resilient systems are able to execute required mission parameters despite a hostile cyber-threat environment.

Image source: adapted from Vugrin et al., 2013

ADDSec Machine Learning

Machine learning algorithms are deployed to each host

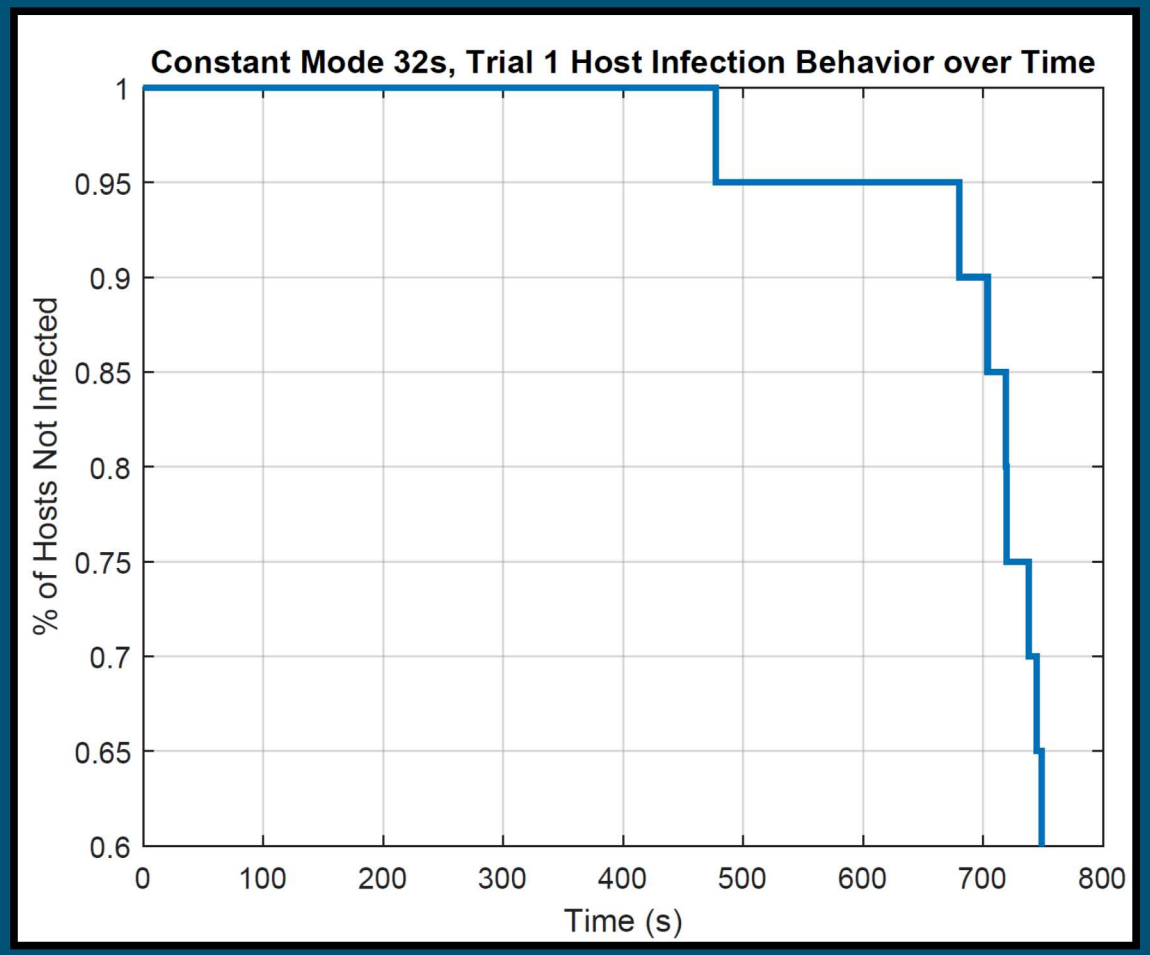
Features extracted from logs on each host:

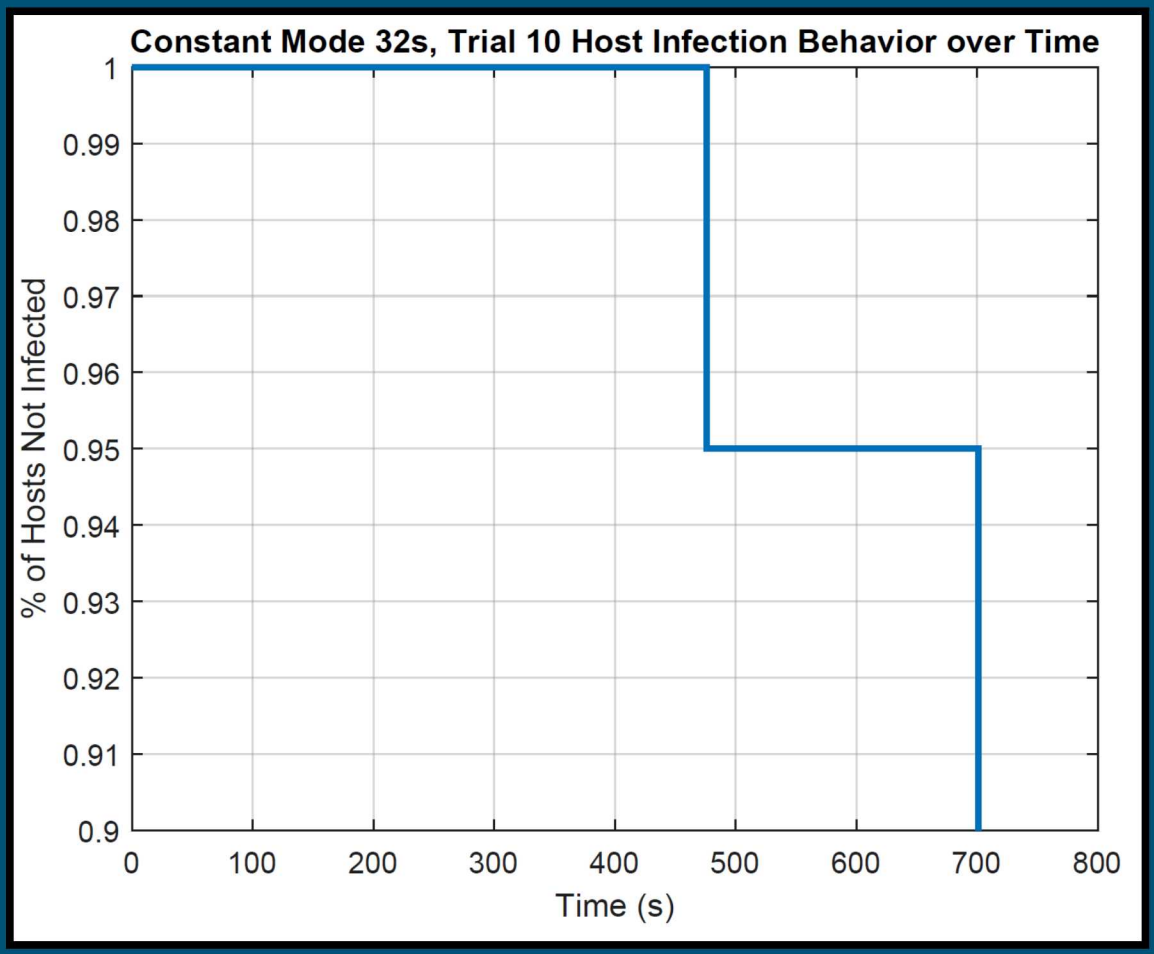
- System status and performance statistics
- System call stack
- Packet capture, Bro network analytics

Classification is performed by an ensemble of techniques (primarily decision trees)

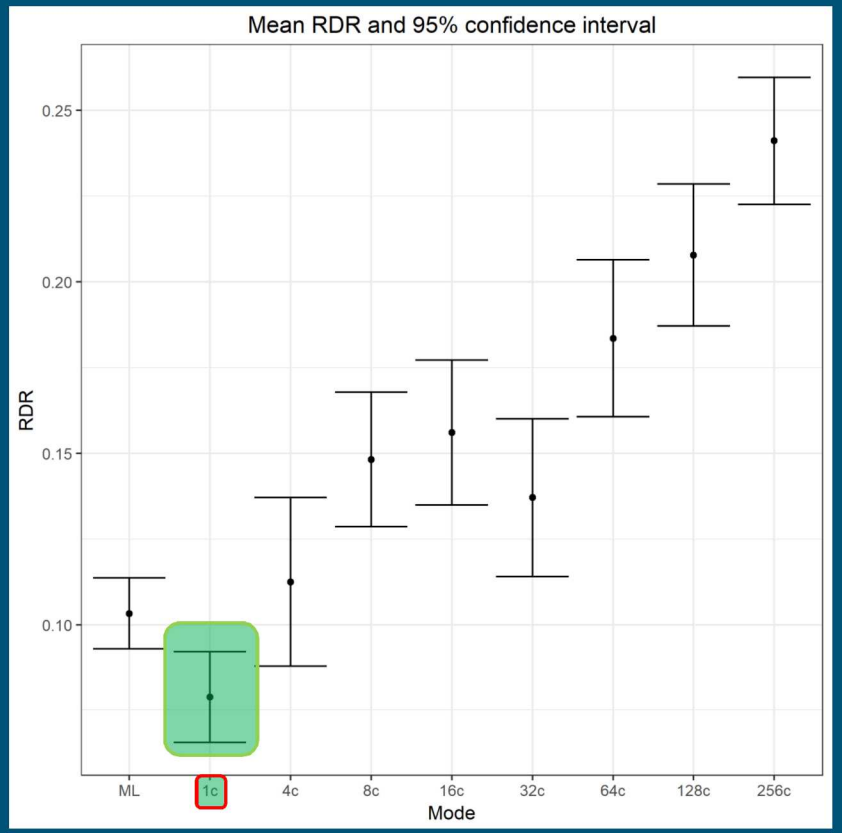
When the machine learning is first turned on, a baseline is taken. The feature set is periodically compared against a baseline and if an alert is triggered, a signal is sent to the controller to undergo randomization.

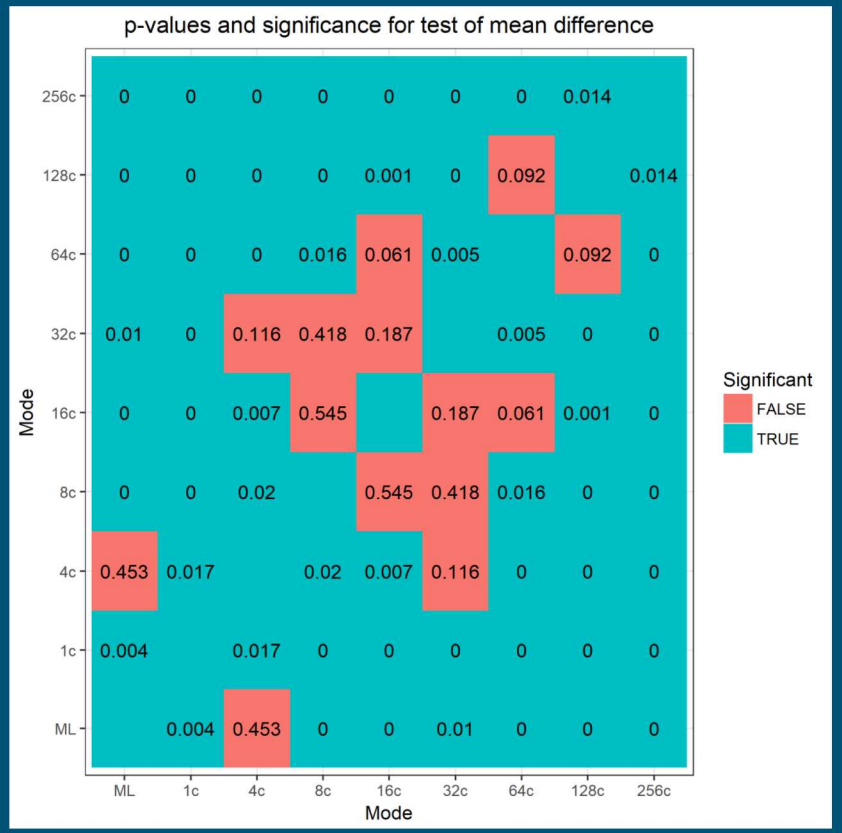
ADDSec Exhibits Stochastic Behavior

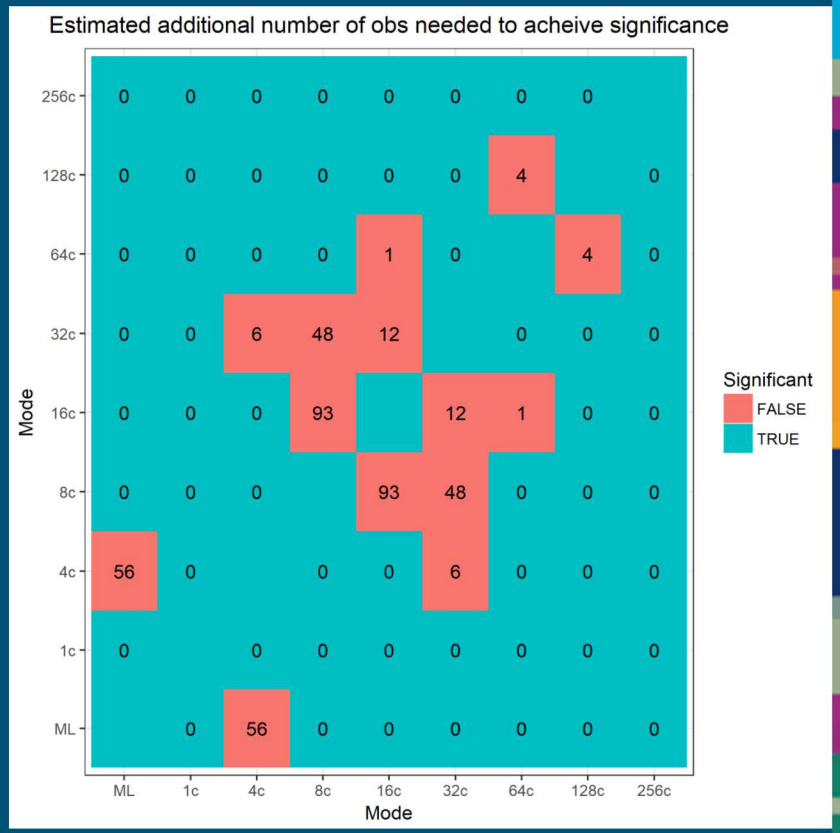




Testing for Significant Differences in RDR







Lessons Learned and Future Experiments

- Pre-processing took substantial effort
  - Automated many processes compared to initial ADDSec analysis
- ADDSec behavior stochastic, needed to collect more data to see more clear trend
  - Difference-in-mean analysis useful for understanding results and if more data needed
  - Gained insight into how to best improve ADDSec behavior:
    - For a predictable scan, randomize among IP ranges that have already been scanned or are not initially scanned.
- Significant effort spent on debugging experiment, determining good data collection strategy and selecting metrics
  - Emulation requires more resources than simulation – deploy experiments on bigger cluster
  - VM resources need to be tuned so that machine learning buffers do not cause crashes
  - Future experiments could be automated with time-based scripts – or port experiment to Firewheel which has time triggers