ARIZONA STATE UNIVERSITY

MTD analysis and evaluation framework in SDN (MASON)

Presented by

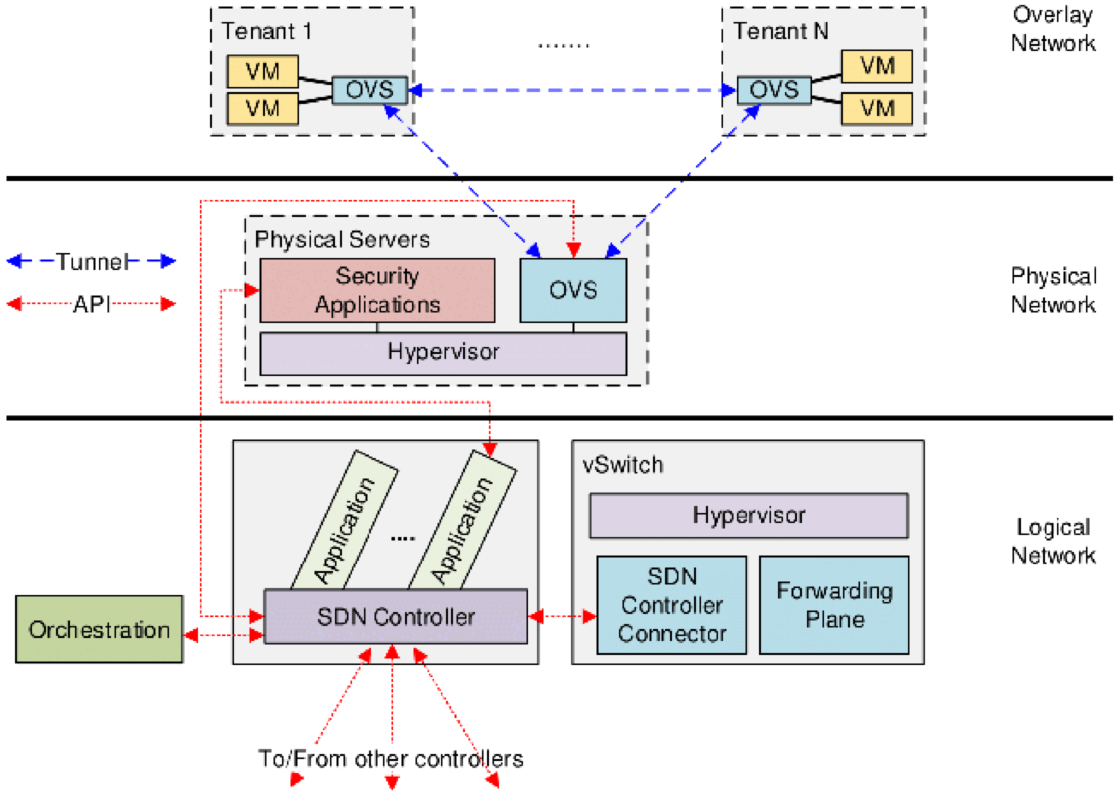
Ankur Chowdhary

SDNNFVSec 2018, Tempe, AZ (03/21/2018)

Index

- Problem Statement
- SDN based MTD
- Threat Model
- MASON Framework
- Experimental Analysis
- Conclusion

Multi-Tenant Cloud Network



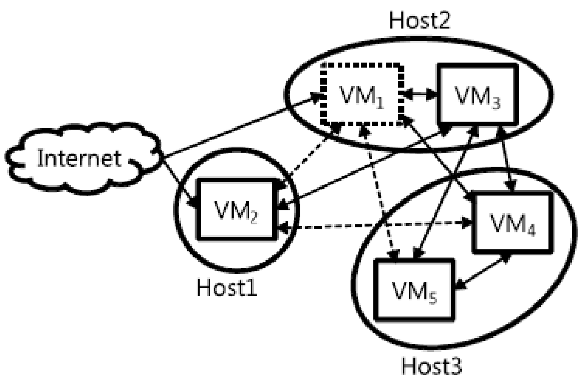
Problem Statement

- Static nature of cloud is bad. Attackers can perform reconnaissance, identify potential vulnerabilities and choose best time to attack.
- Lack of correlation mechanism between static vulnerabilities and dynamic traffic information in the network.
- Limited Security Budget.

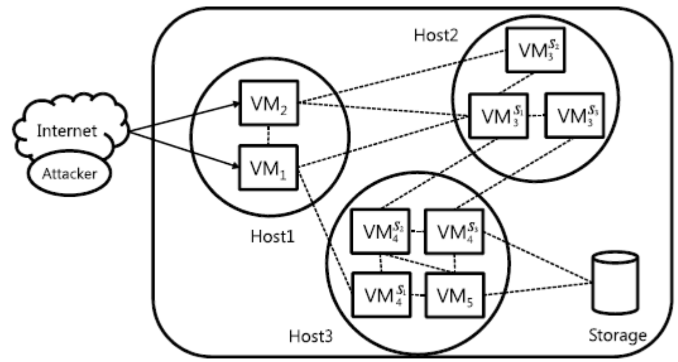
Moving Target Defense

Moving Target Defense (MTD) is the concept of controlling change across multiple system dimensions in order to increase uncertainty and apparent complexity for attackers, reduce their window of opportunity and increase the costs of their probing and attack efforts.

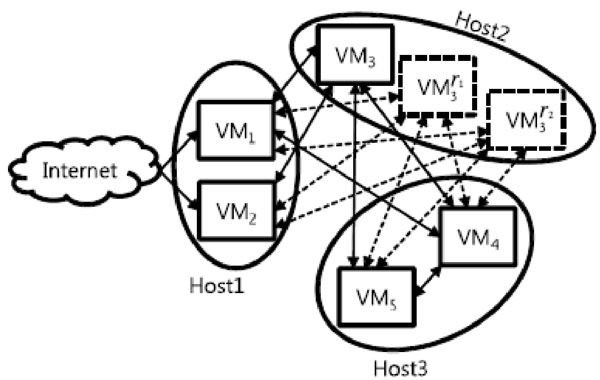
MTD Techniques



Shuffle



Diversity



Redundancy

SDN based Moving Target Defense

- Software Defined Networking (SDN) provides centralized command and control (C&C) for a cloud network.
- The scope of pro-actively making changes to the cloud network makes SDN as an ideal candidate for deploying MTD.

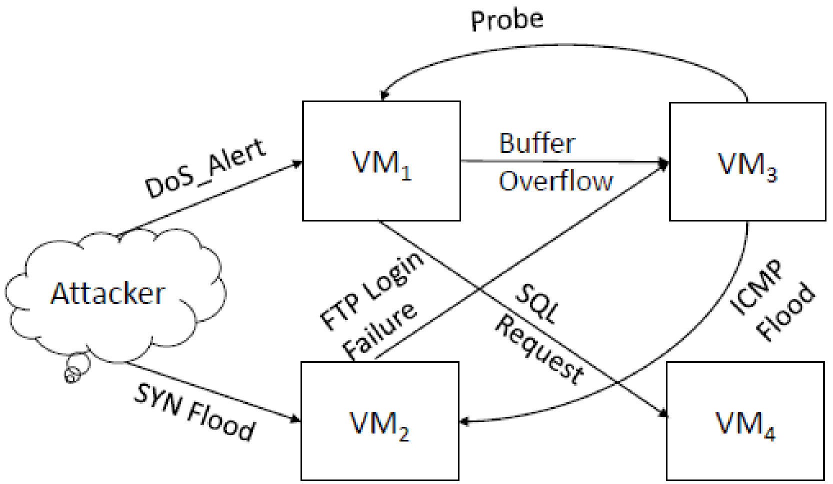
Contribution

Problem	MASON Solution
Static resources with vulnerabilities	Moving Target Defense using SDN
Lack of Correlation between vulnerabilities and IDS alerts	Threat scoring algorithm based on Page Rank
Limited Security Budget	Prioritization of high services with high threat score

Related Work

Paper	Approach	Comments
"Kampanakis et Al" **SDN-based solutions for Moving Target Defense network protection**	Send random payload or random network traffic in response to the attacker reconnaissance request	Network mapping and reconnaissance protection, service version and OS hiding
"Debroy et Al" **Frequency-Minimal Moving Target Defense using Software-Defined Networking**	Frequency minimization and consequent location selection of target movement across heterogeneous virtual machines based on attack probability	VM migration using factors such as network bandwidth, VM capacity, VM reputation, etc.
"Jafarian et Al" **OpenFlow Random Host Mutation: Transparent Moving Target Defense using Software Defined Networking**	Each host associated with unused network address range based on specific requirement.	vIPs of all hosts in subnet during *T* must be less than the aggregate size of all ranges. Ranges must be assigned to subnets proportionate to their total required mutation rate.
Zhao et al **An SDN-Based Fingerprint Hopping Method to Prevent Fingerprinting Attacks**	Signature based attack detection. FPH tries to change some attributes of the packets of outgoing traffic to defense fingerprinting attackers.	Signaling game, where Perfect Bayesian Equilibrium (PBE) is used to predict the outcome of the game.

Threat Model



a) Intrusion Events reported by IDS

VM	Vulnerability	CVSS Score
VM$_1$	Memory Consumption	7.0
VM$_2$	Buffer Overflow	8.9
VM$_3$	Integer Overflow	8.6
VM$_4$	SQL Injection	10.0

b) Vulnerabilities reported by scanner

Event Correlation

- Represent IDS events and vulnerabilities as graph, $G = \{S, A, \rightarrow, I\}$.
- States represent vulnerability states (buffer overflow). Edges are attackers actions, e.g. SQL Injection.
- Attack probability depends on normalized IDS severity of all IDS alert events, s.t. $\forall s \in S, \exists a \in A; \sum a = 1$.

Attack Propagation

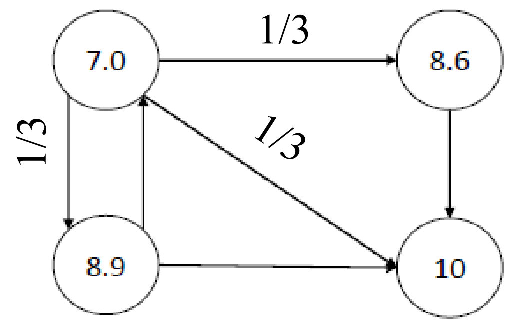
- We consider attacker as a random surfer in network similar to page rank model.

- The probability of attacker exploiting a vulnerability x_i is given by equation below.

$$x_i = \frac{1-d}{N} + d \sum_{j \in In(i)} \frac{x_j}{|Out(j)|}$$

- |Out(j)| shows the number of outgoing connections from vulnerable service, d is damping factor to show randomness of attacker's action.

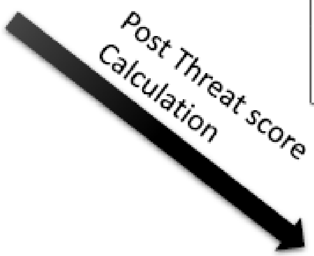
Threat Score



TS = [7.0 8.9 8.6 10]

$$A = \begin{bmatrix} 0 & 1/2 & 0 & 1/4 \\ 1/3 & 0 & 0 & 1/4 \\ 1/3 & 1/2 & 0 & 1/4 \\ 1/3 & 0 & 1 & 1/4 \end{bmatrix}$$

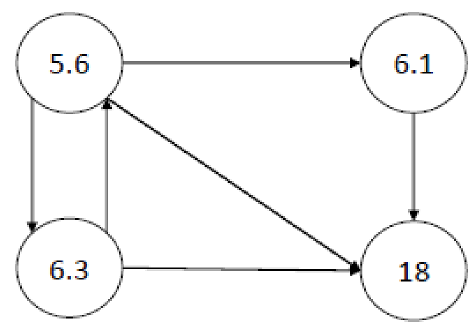
d = 0.10



$$\bar{a}_{ij} = \begin{cases} \frac{1}{L(j)} & \text{if there is a link from node } j \text{ to node } i, \\ \frac{1}{N} & \text{if node } j \text{ is a dangling node,} \\ 0 & \text{otherwise,} \end{cases}$$

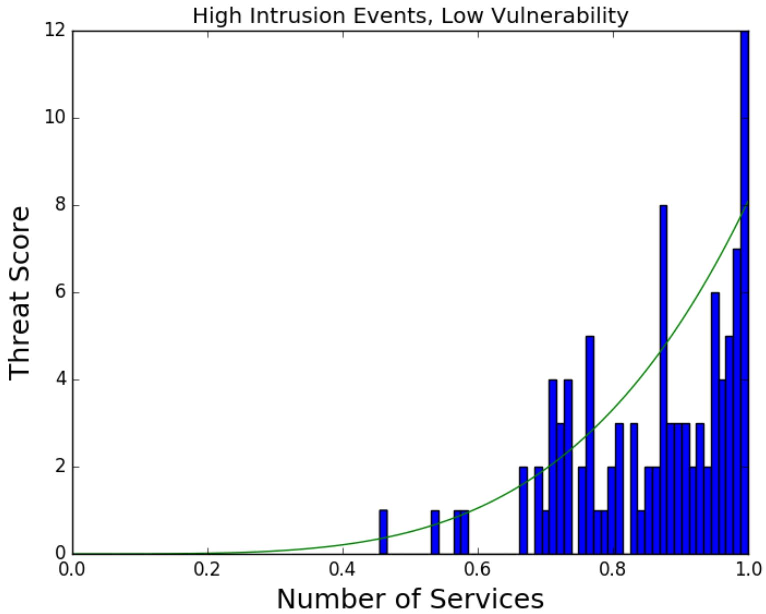
iteration	x_A	x_B	x_C	x_D
0	1	1	1	1
1	0.75	0.5833	0.5833	2.0833
2	0.8125	0.7708	0.7708	1.6458
3	0.7969	0.6823	0.6823	1.8385
4	0.8008	0.7253	0.7253	1.7487
5	0.7998	0.7041	0.7041	1.7920

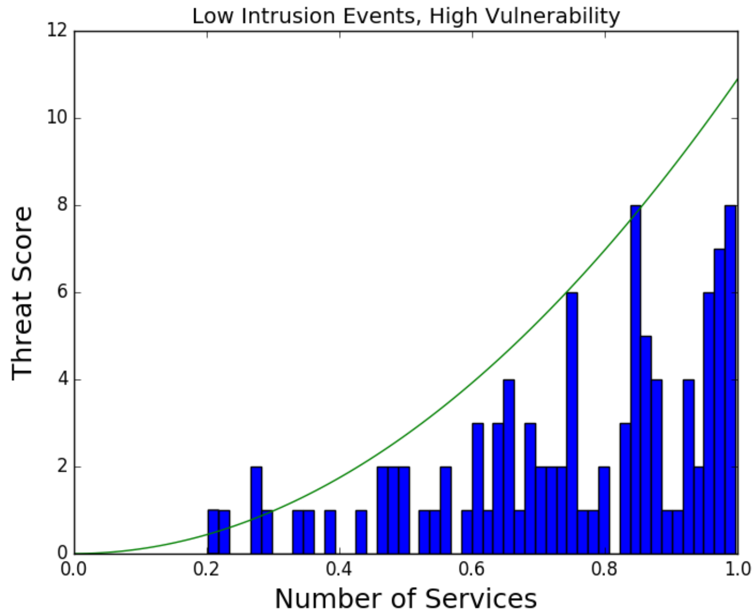
TS = TS * $(x_A, x_B, x_C, x_D)^T$



TS = [5.6 6.3 6.1 18]

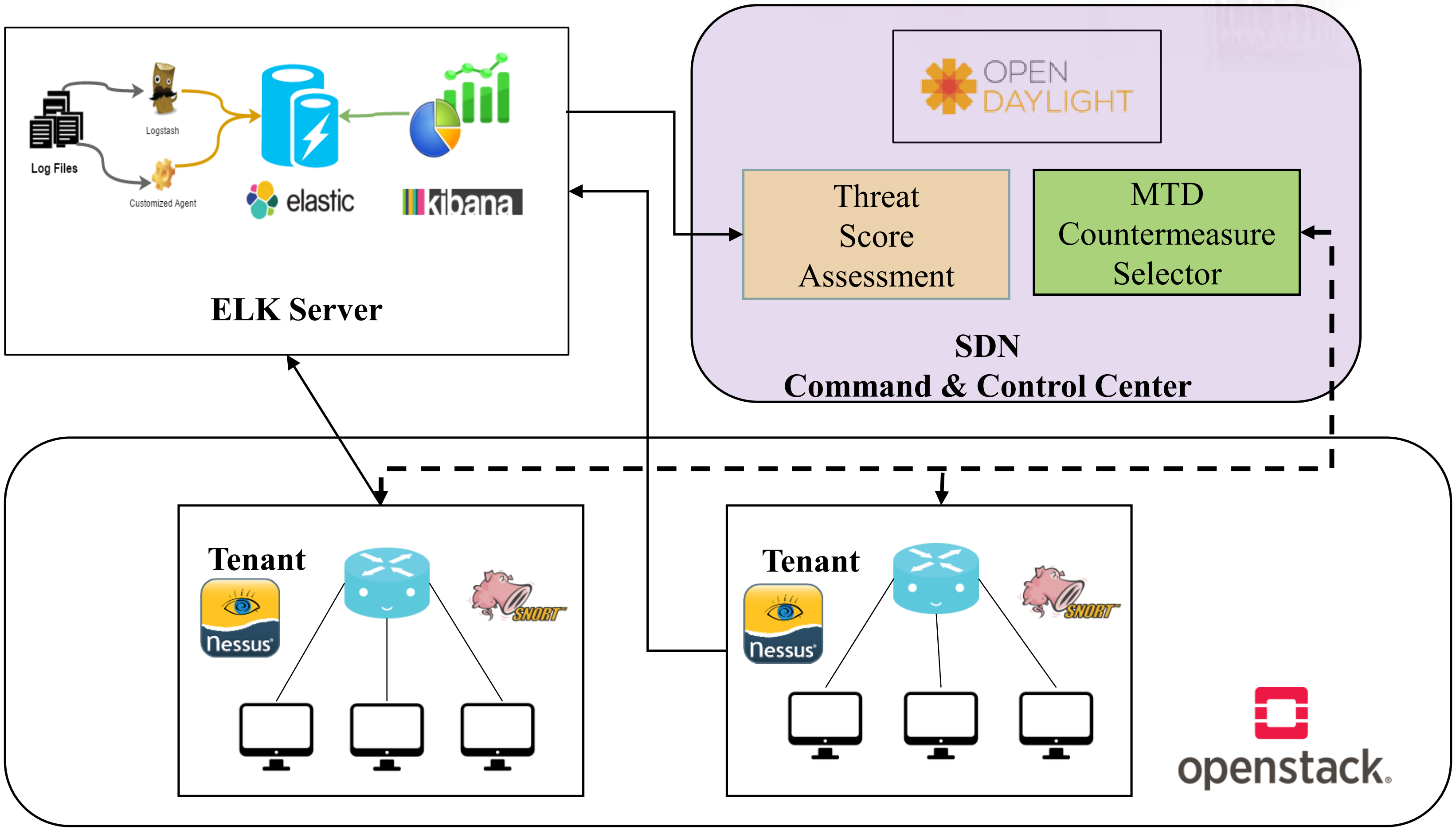
Threat Score





Arizona State University

MASON Architecture



Network Threat Scoring Algorithm

Algorithm 1: NETWORK-THREAT-SCORING

Input: $G = \{N, E\}$, d=0.1

Output: Emit- TS(N) : Threat Score on Algo. Termination

A: Link Matrix

$a_{ij} \in A \leftarrow \frac{1}{L(j)}$: For link from node j to i

$a_{ij} \leftarrow \frac{1}{N}$: For dangling node j

$a_{ij} \leftarrow 0$: Otherwise

$x \leftarrow (1, 1, 1, ..)$: Probability Matrix

$k \leftarrow 0$

$\epsilon \leftarrow 1 \times 10^{-3}$: Algo. stopping criteria

TS(N) : Initial threat score of nodes

while $|x_{k+1} - x_k| \leq \epsilon$ **do**

  $x_{k+1} \leftarrow \frac{1-d}{N} + dA^T x_k$

  $k \leftarrow k + 1$

return x

$TS = TS \times x^T$

Experimental Analysis

Threat Score vs Services

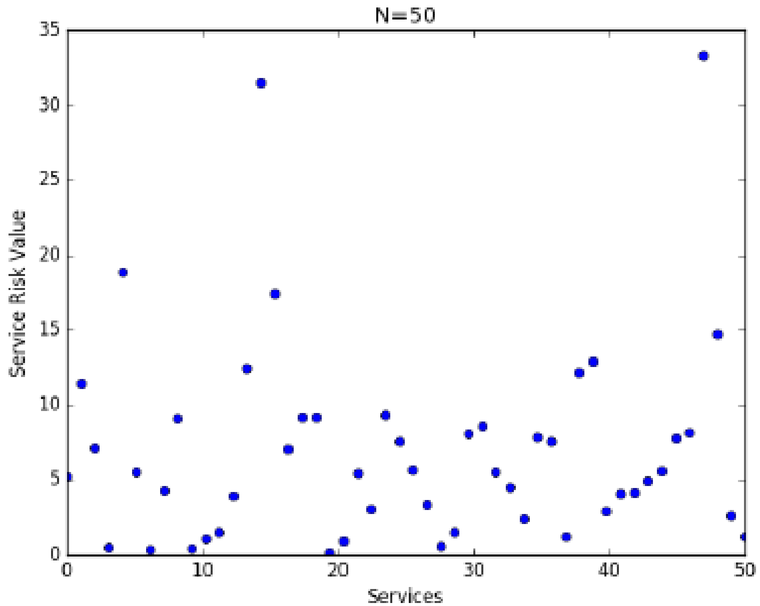


Figure 5: Threat Score vs N=50 Number of Services

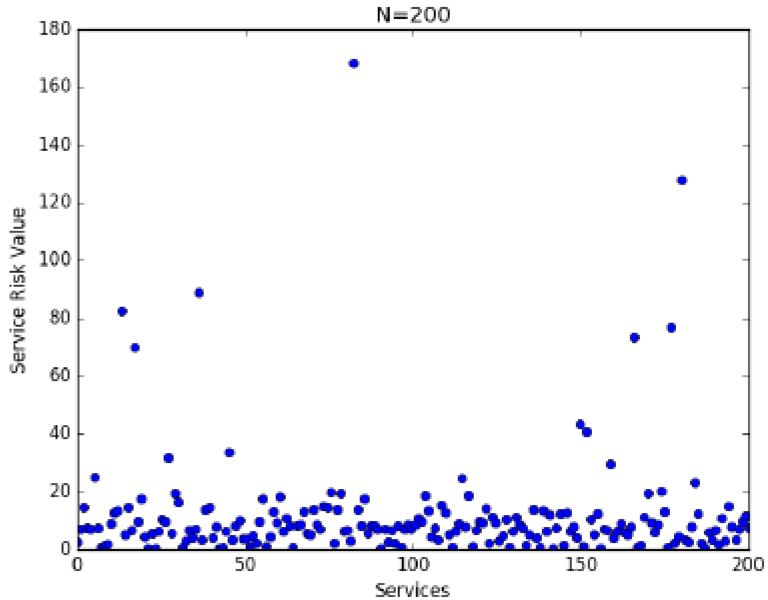


Figure 6: Threat Score vs N=200 Number of Services

Threat Score vs Services

- Very few services such as telnet, rlogin had very high threat score.
- Services with high network centrality had high threat score.
- Deployment of MTD countermeasure can be prioritized in order of decreasing threat score.

Threat Score vs MTD Port Hopping

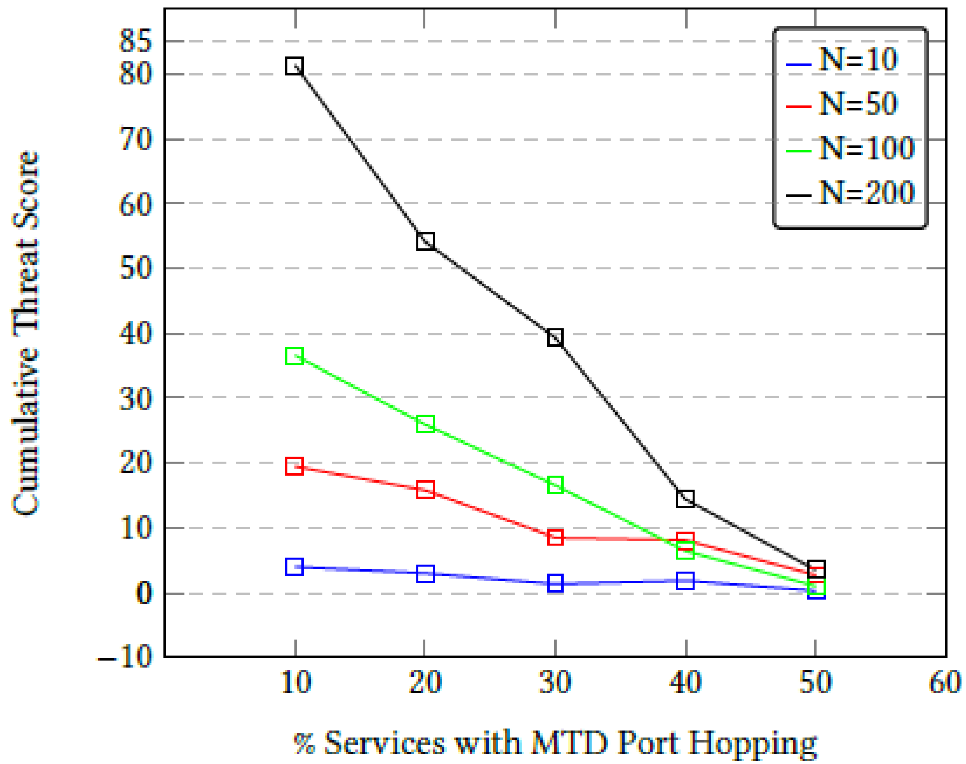


Figure 7: Threat Score vs MTD port hopping

Conclusion and Future Work

- Threat scoring based on Page Rank algorithm to select services with high threat score for MTD.
- Marked reduction in threat score of network by prioritizing MTD deployment.
- Approach is limited to signature based detection methods.

Questions ??