University of Missouri

Frequency Minimal Moving Target Defense Using Software-Defined Networking

Allen Stage and Vladimir Georgiev

Mentors: Prof. Prasad Calyam, Dr. Saptarshi Debroy, Minh Nguyen

Acknowledgements

geni

THOMSON REUTERS

This material is based upon work supported by the National Science Foundation under Award No. CNS-1359125, Thomson Reuters and University of Missouri. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or Thomson Reuters.

Threats to Cloud Security

Cloud systems can be vulnerable to a variety of threats:

- Information Leakage
  - Cause: Eavesdropping, Traffic Interception
  - Effect: Loss of confidentiality
- Integration Violation
  - Cause: Intercept/Alter ,Repudiation
  - Effect: Loss of integrity
- Denial of Service
  - Cause: Trojan Horse, Resource Exhaustion
  - Effect :Loss of Availability
- Illegitimate Use
  - Cause: Spoofing, theft
  - Effect: Improper Authentication

Top Cloud Computing Threats in 2013

1. Data Breaches	**2. Data Loss**	3. Account Hijacking	4.Insecure APIs
5. Denial of Service	**6. Malicious Insiders**	7.Abuse of Cloud Services	8.Insufficient Due Diligence

Denial of Service and Loss of Availability

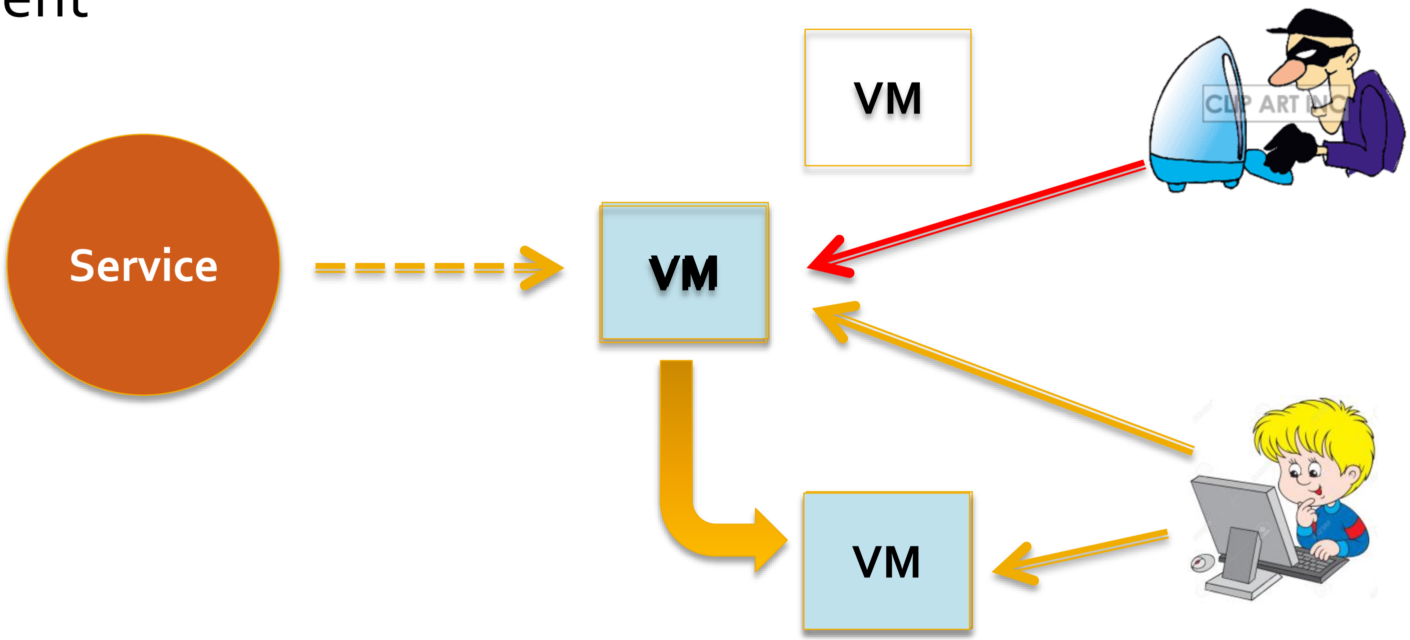
- LOA
  - Loss of Availability
- DOS
  - Denial of service
  - Simple to execute
  - Attacker bombards a server with requests, and render it completely useless for other users

Traditional Security Strategies

- Cryptographic strategies against LOA
  - Proof-of-Retrievability (POR) [1]
  - Proof of Data Possession (PDP) [2][3][4][5]
- Strategies against DDOS
  - Router filtering [6][7]
  - Instrument prevention system (IPS) [8][9][10]

What is Moving Target Defense?

- Moving target defense (MTD) is the concept of controlling change across multiple system dimensions by moving around VMs hosting services
- MTD focuses on enabling safe operation in a compromised environment, rather than trying to create a perfectly secure environment



Why MTD for cloud security?

- Improves resilience through randomization, helps achieve cyber defense goals
  - Increased cost to attacker
  - Decreased knowledge of whether or not attack was successful
  - Increased chance of attacker detection
- Contains *proactive* (preventive) and *reactive* (cure) defense to prevent attacks
- Intelligent proactive and reactive strategies can help tackle LOA attacks!

Related Work on MTD for Cloud Security

Related work	Strengths	Limitations
[11]	Shuffling static IP addresses of attacked VMs	Only reactive strategy
[12]	Moving proxies to application servers to thwart attack	Attacker can realize defense strategy in place
[13]	Proactive VM migration using attack traffic signature	Too reliant on accuracy of signature detection
[14]	Multiple VMs host same service, users are only redirected	Not really MTD, limited cost-effectiveness
[15]	Attackers are marginalized within a small pool of decoy VMs	Does not guarantee 100% regular user redirection

Our Research Goals

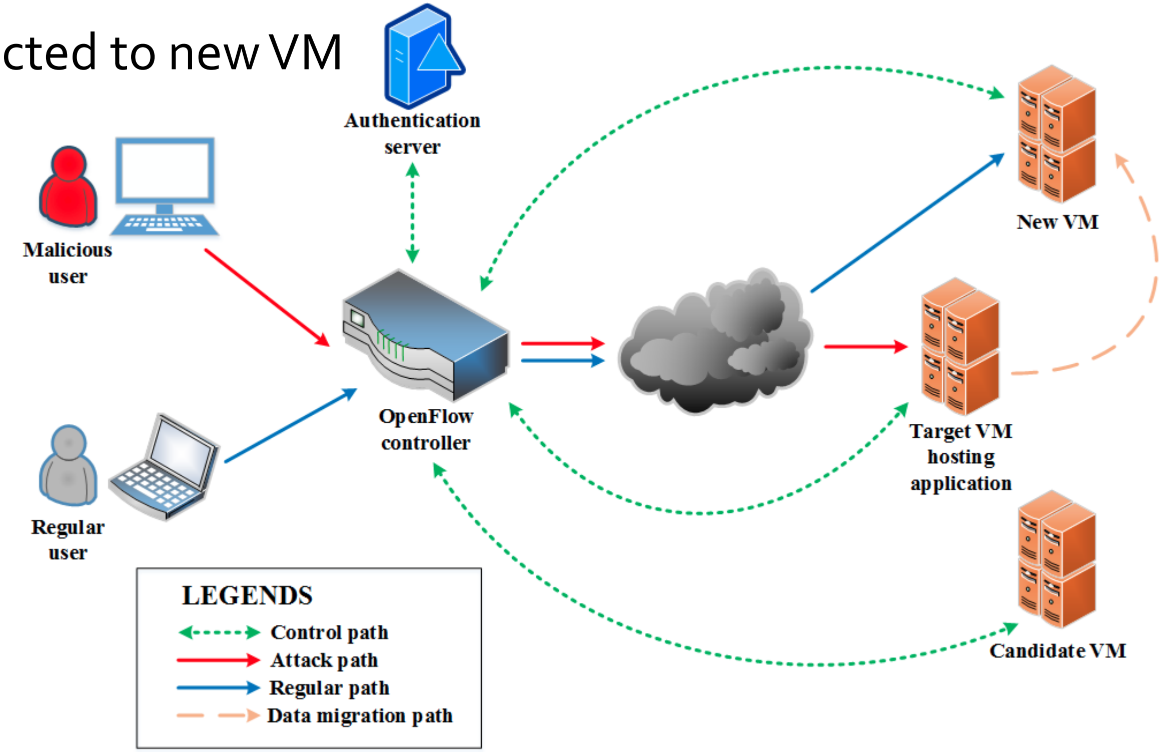
- Both proactive and reactive movement strategies
- Optimal cost effective migration strategy
- Trade-off between cost of movement and difficulty for attacker to guess
- Attacker should not know about the movement and keep targeting the old VM

Our FM-MTD Novelty

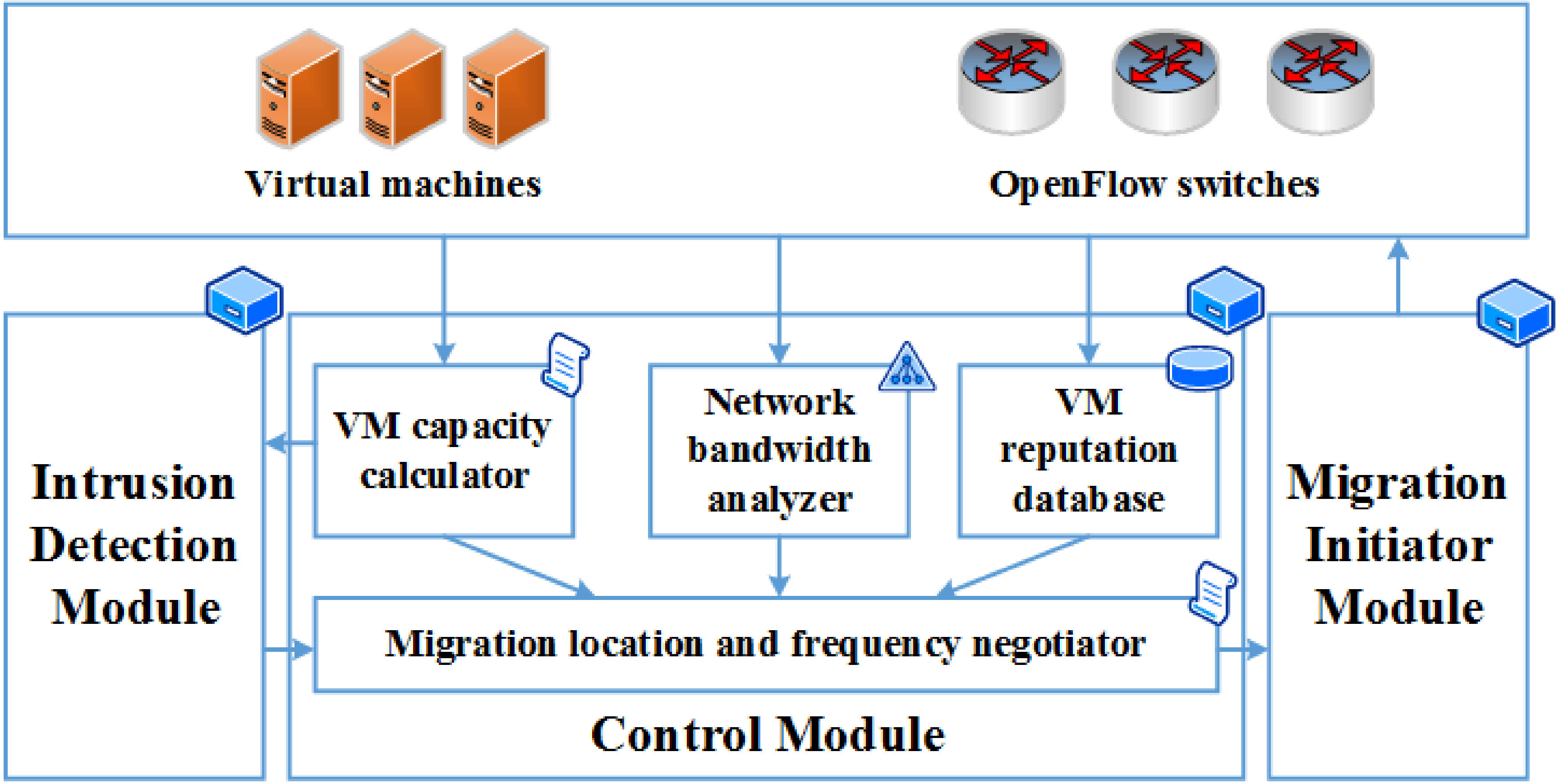
- Our SDN-enabled migration scheme performs dynamic VM migration
  - Whereas, existing works resorts to IP address shuffling
- Our scheme is both proactive and reactive
  - Whereas, existing works are purely reactive
- Our scheme is adaptive to attack probability and attack budget
  - Whereas, existing use migration frequency that is static
- Our scheme considers heterogeneous VM pool
  - Whereas, existing works assume a homogeneous VM pool

MTD System Model

- Malicious and regular users accessing the services hosted by a target VM
- Authentication server to authorize users
- Open flow controller to detect attack, run MTD logic, and perform migration
  - Only regular users redirected to new VM



MTD Controller Architecture



Three Big Questions

- Where to move?
  - Finding the optimal candidate VM to migrate
  - Identifying the most pertinent VM selection factors
  - Periodic/on-demand information collection
  - Finding the factors' relative importance to create migration logic
- When to move?
  - Finding the optimal frequency of movement
  - Not too frequent as migration incurs cost, and not too seldom as increases probability of getting attacked
- How to move?
  - Mostly pertains to implementation issues
  - Proactive/reactive migration execution
  - Runtime migration or file copy
  - Redirection of regular users

Optimal Migration Frequency

- Ideal frequency should be such that it is not too frequent, while not being too infrequent
- Too frequent
  - can waste valuable network resources
- Too infrequent
  - makes VM more vulnerable

Movement costs resources, just like moving houses costs time and money

Attack Budget and Probability

- The optimization can be formulated as

$$maximize(T_m)$$

$$T_m \leq \text{ cyberattack inter-arrival time}$$

- Assume the random variable representing the attack inter-arrival time be **z** which is the sum of two independent and random variables for Attacked and Idle periods x and y, respectively.
- The distribution of attack interval **z** is obtained by:

$$\begin{aligned} f_Z(z) &= f_X(x) * f_Y(y) \\ &= \int_{-\infty}^{+\infty} f_X(z-y) f_Y(y) dy \\ &= \begin{cases} \frac{\lambda_a \mu_i [e^{-\lambda_a z} - e^{-\mu_i z}]}{(\lambda_a - \mu_i)} & \forall\ \lambda_a \neq \mu_i \\ \lambda_a^2 z e^{-\lambda_a z} & \text{otherwise} \end{cases} \end{aligned}$$

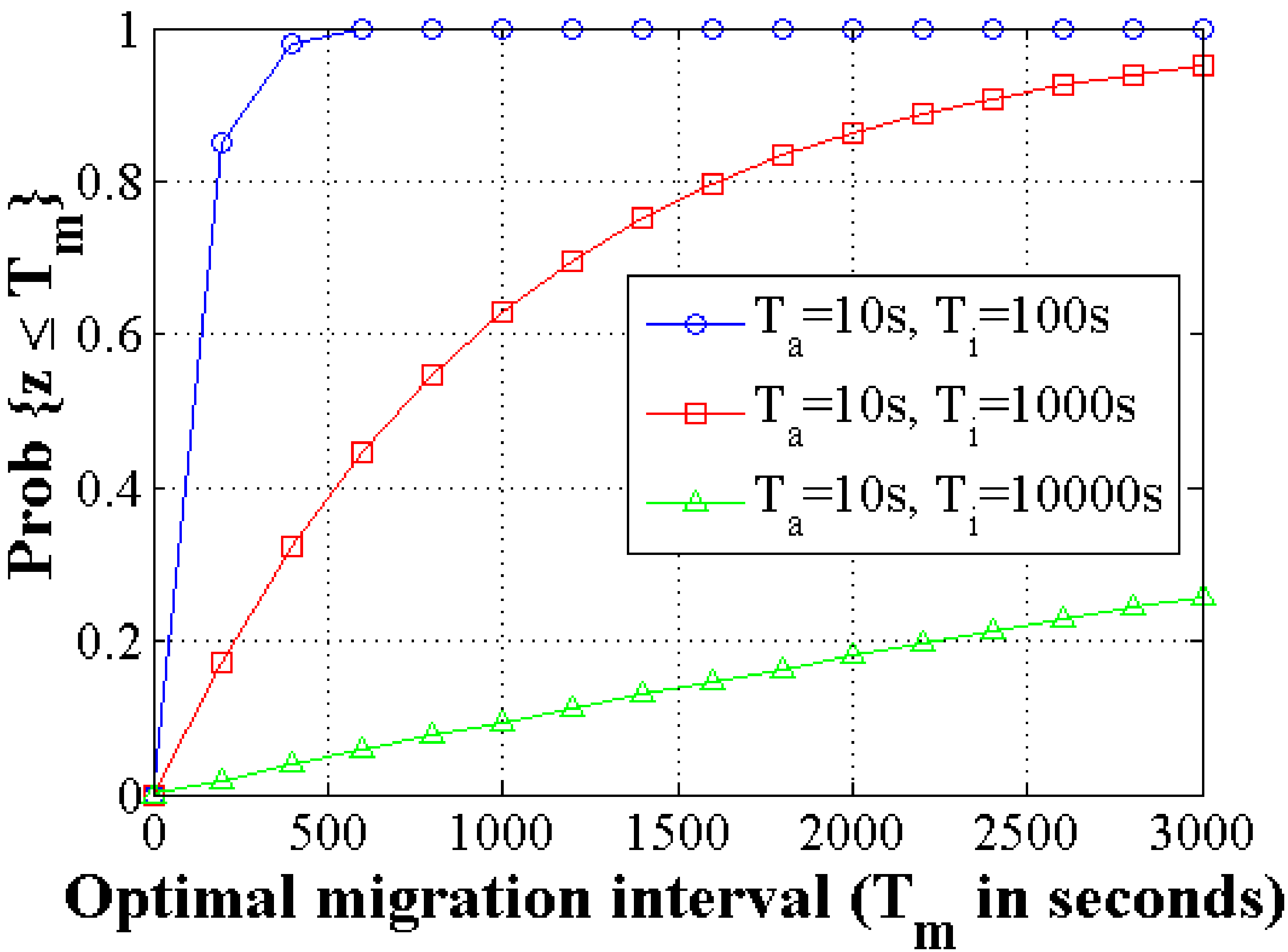
Attack Budget and Probability (2)

- To quantify optimal T_m, calculate probability of VM getting attacked before migration

$$
\begin{aligned}
&\text{Prob}\{\text{VM getting attacked before migration}\} \\
&= \text{Prob}\{\text{z} \le T_m\} \qquad \text{(VM attack being memoryless)} \\
&= \int_{-\infty}^{T_m} f_Z(z)dz \\
&= \begin{cases} \int_0^{T_m} \frac{\lambda_a \mu_i [e^{-\lambda_a z} - e^{-\mu_i z}]}{(\lambda_a - \mu_i)} dz & \forall\ \lambda_a \ne \mu_i \\ \int_0^{T_m} \lambda_a^2 z e^{-\lambda_a z} dz & \text{otherwise} \end{cases} \\
&= \begin{cases} \frac{\mu_i (e^{-\lambda_a T_m} - 1) + \lambda_a (1 - e^{-\mu_i T_m})}{\lambda_a - \mu_i} & \forall\ \lambda_a \ne \mu_i \\ 1 - e^{-\lambda_a T_m}(\lambda_a T_m + 1) & \text{otherwise} \end{cases}
\end{aligned}
$$

$Lambda_a$ is representative of attack period
Mu_i is representative of idle period

Migration interval (T_m) optimization for different attack budgets



A visual representation of the equation slides, with many movement frequencies in a graph

Ideal Migration Location

- VM selection factors:
  - Capacity: New VM should have enough resources (compute/storage)
  - Bandwdith: New VM should not be too far to cause extended service interruptions
  - Reputation: New VM should not be prone to attack or have prior history of getting attacked
- Selection criteria

$$maximize(\mathcal{S}_p^v)$$

where $$\mathcal{S}_p^v = w_c \times C_p + w_b \times B_p^v + w_r \times R_p^j$$

Reputation in Depth

- We argue that the previous history of a VM in terms of instances of cyber attacks is a critical factor in deciding the suitability for selection
  - *Instances of successful attacks (alpha)*
  - *Instances of unsuccessful attacks (beta)*
  - *Instances of attack-free status (gamma)*
- Cumulative fair reputation model

$$R_p^j = 1 - \frac{\alpha_p^j + \frac{\beta_p^j}{\beta_p^j + \gamma_p^j}}{\alpha_p^j + \beta_p^j + \gamma_p^j} \quad \forall\ p \in V$$

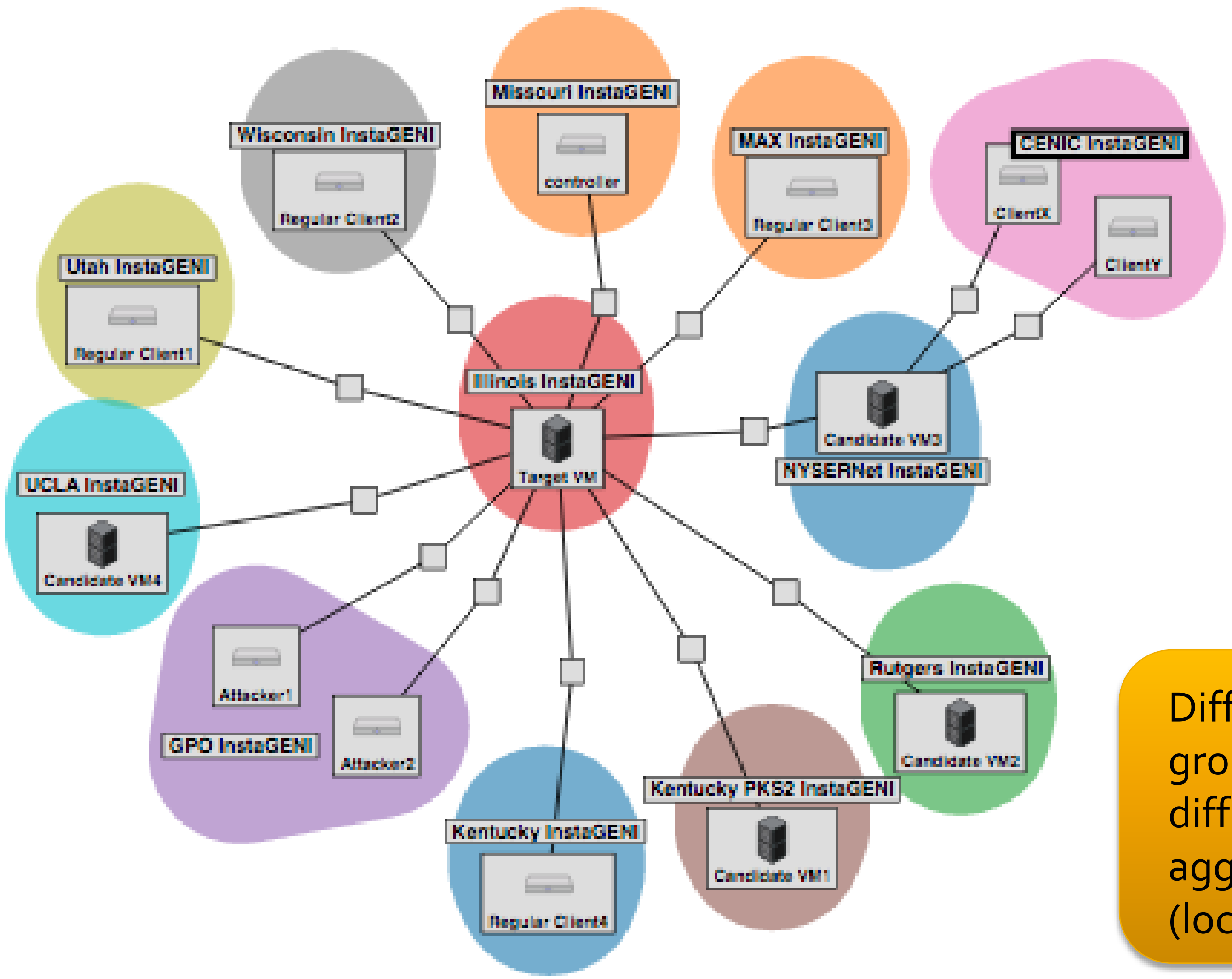
Performance Evaluation

- Target Application – Just-in-time news feeds
- Using a software-defined networking controller we developed
  - Contains python and shell scripts that we have written to execute the movement modules
- Scripts will move our application to a new VM

Experiment

- Setup on testbed consists of the following components
  - One target VM at Illinois rack hosting the target application
  - Four non-malicious clients at four different locations
  - Two attackers simulating regular client behavior
  - Up to 30 candidate VM's at different locations simulating varied scenarios
  - Controller with software components of control module

Performance Evaluation



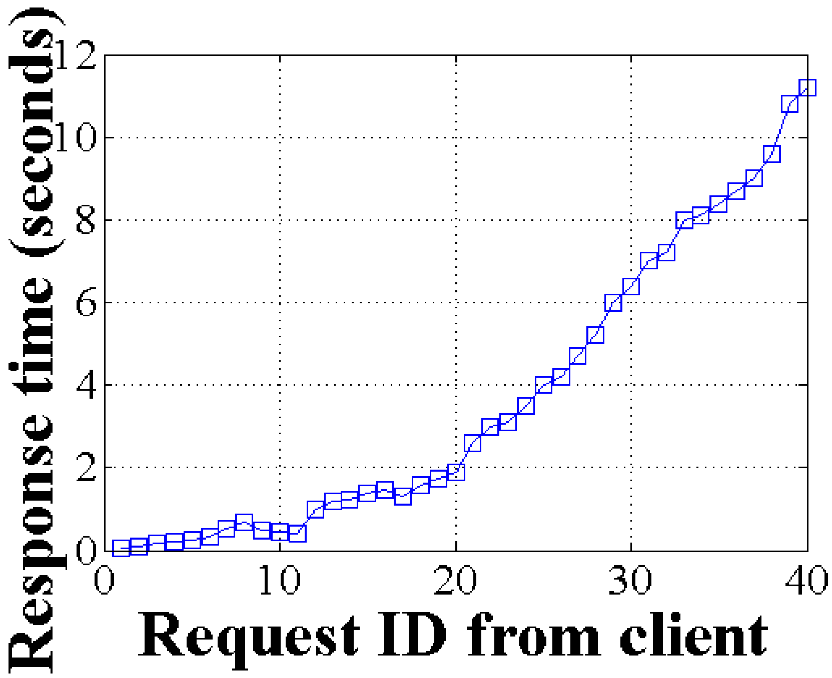
Different color groups represent different aggregates (locations)

Cyber attack Impact

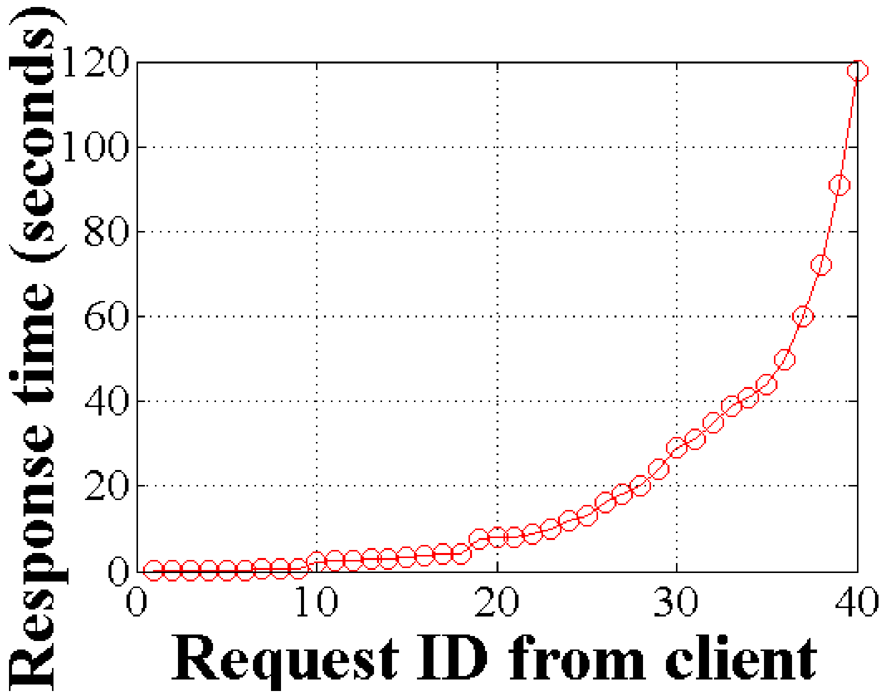
Impact of cyber attack on requests from client4

Notice the trend? (Hint: the axis matter)

1 attacker

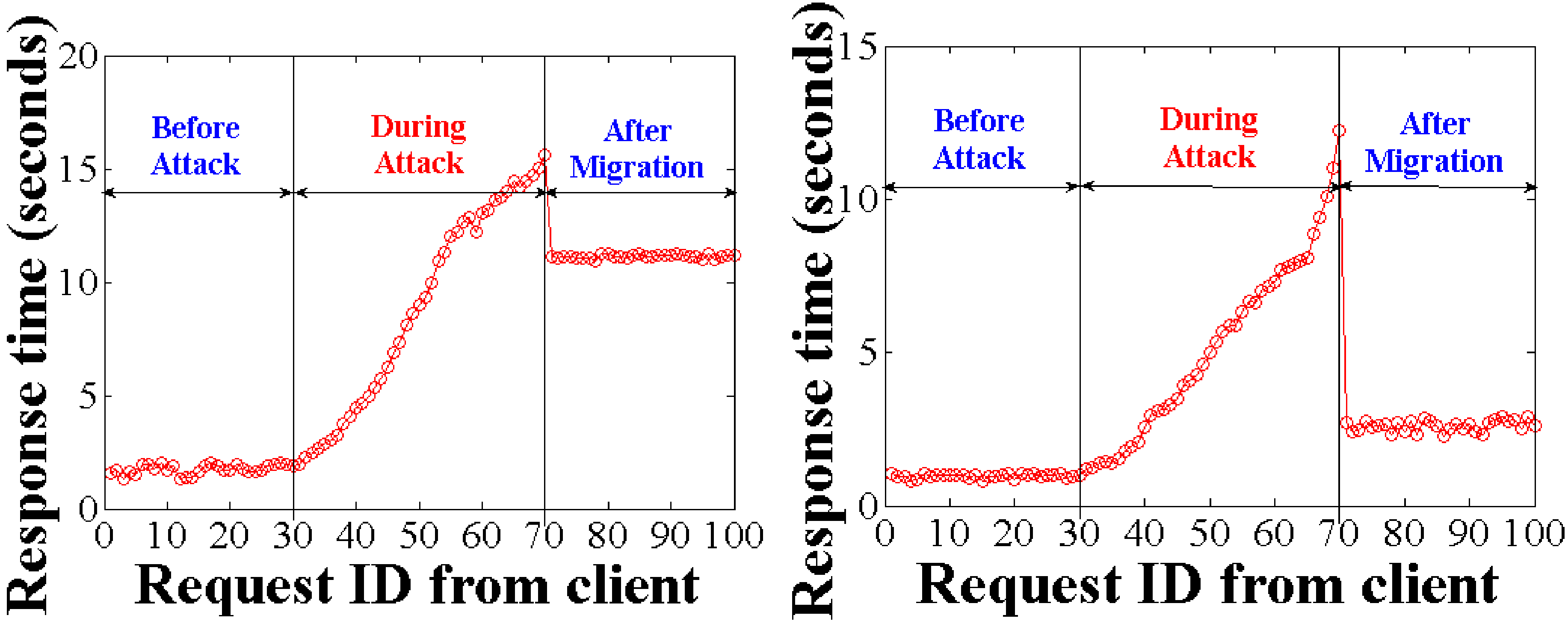


2 attackers



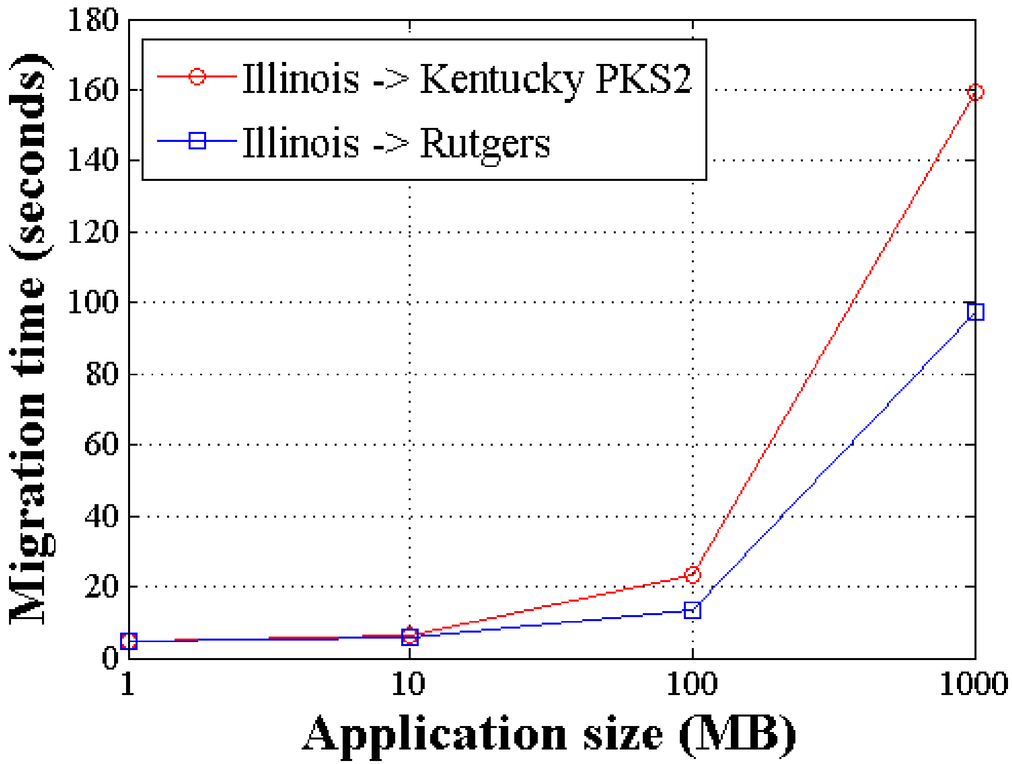
Impact of Location Selection

- Process of selecting ideal frequency minimal candidate VM over static homogenous
- Response time for client4 with a less than ideal VM can lead to service quality improvement, compared to attack, but quite less when compared to ideal, in this case up to a factor of ~4



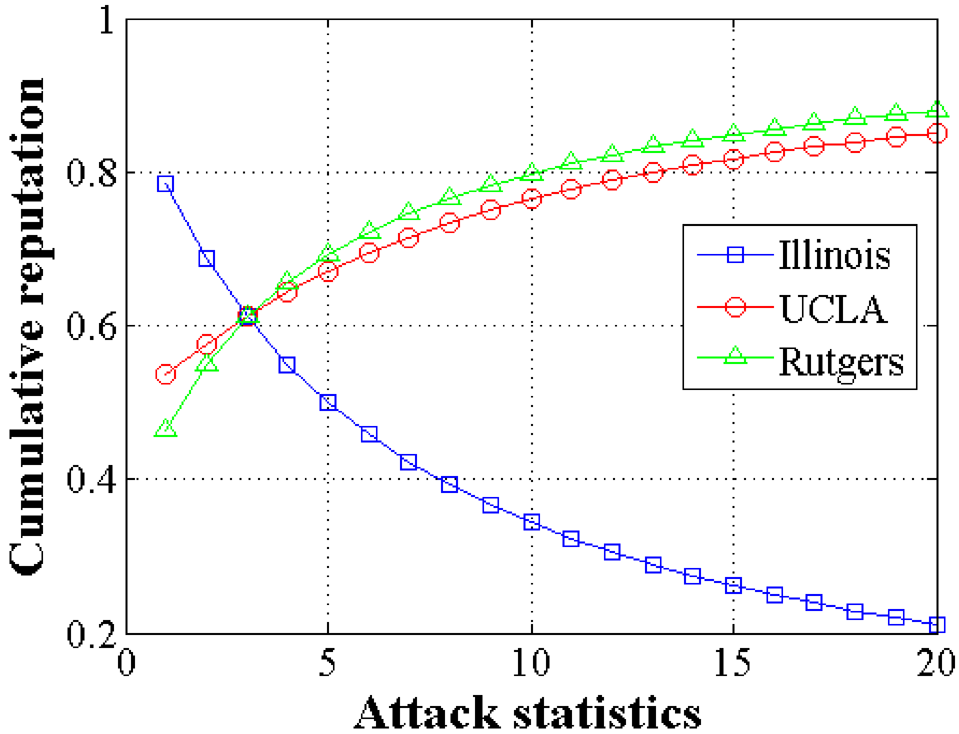
Impact of Bandwidth

- Installed Kentucky PKS2 with similar features as our ideal candidate, the exception being the achievable throughput
- Varying the size of the application
- Increased transfer times affects the service interruption time in the case of an attack



Cumulative Reputation

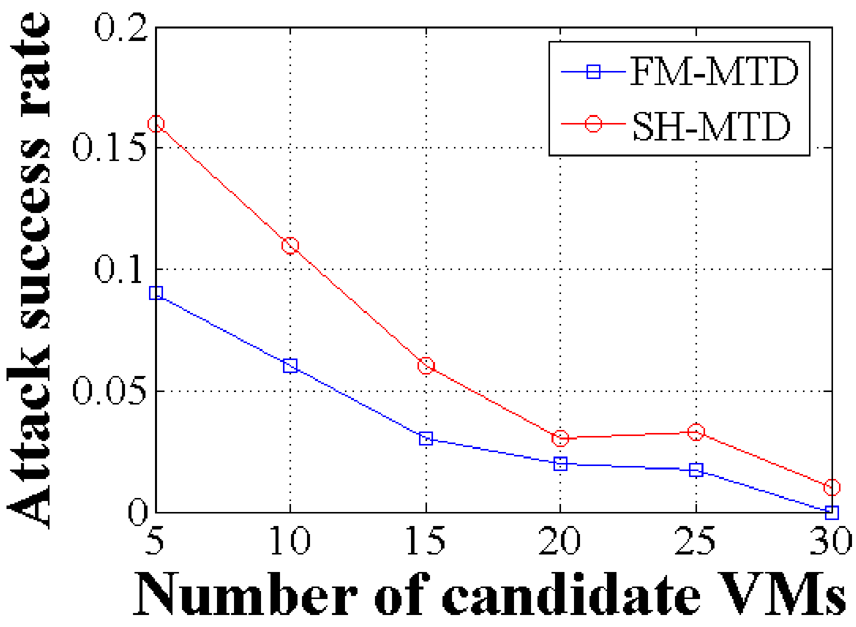
- Illinois is targeted, while hosting
- UCLA is targeted, but not hosting
- Rutgers is not targeted



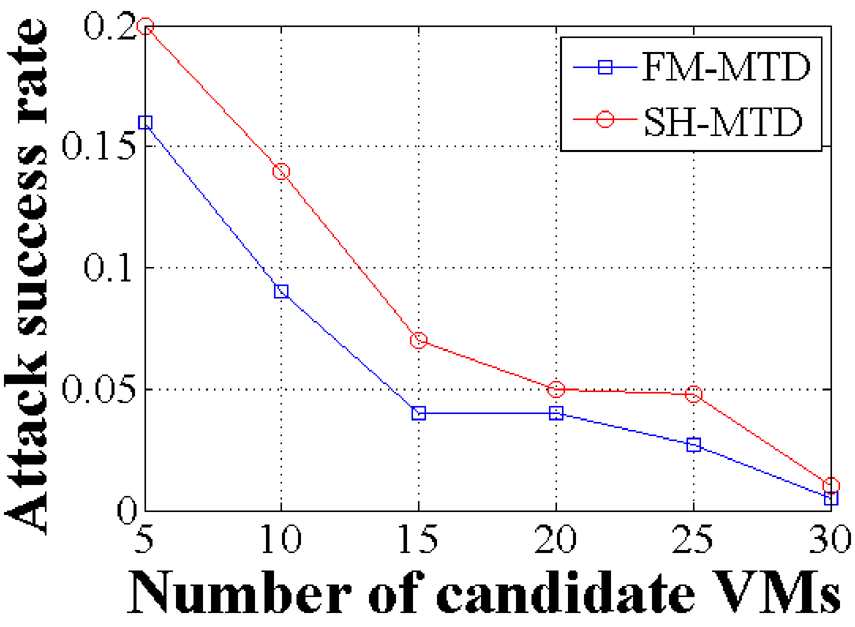
Proactive Migration Performance

- This time proactive is performed, varying the probability of the attack by varying attack budget
- Optimal migration frequency performs better, up to 50% at lower ends
- Success rate sharply decreases with growing number of VM's, as guessing out of 30 versus 5 becomes more difficult

1/10 budget ratio



1/100 budget ratio



Conclusion

- Proactive movement using our ‘when to move’ module is successful in preventing a greater number of attacks
- Reactive movement using our ‘where to move’ module results in a better response time

Further thoughts and future considerations

- Larger amounts of VM's created larger run times in the modules, as would be expected
- A thought on this would be that with a larger number of VM's the attack probability becomes extremely low anyway, as determined by the frequency optimization
- Another thought on this is controller type, as discussed in the next slide

Things we would do different

- We started on DeterLab then switched to GENI
  - Overall, this turned out to be a good thing! But did come at a cost for only having 10 weeks
- Time-management
  - An example is "wasted" time on irrelevant problems (such as with DeterLab node login)
    - These things improved drastically with experience!
- Experiment with controllers other than POX

It is a learning process!

What we learned and takeaway

- LaTex, and other ins and outs of research paper fundamentals
- Presentation giving on a weekly basis, as well as listening skills involved in them
- Many different areas from just our own project!
  - Software-Defined Networking fundamentals
  - Moving Target Defense Fundamentals
  - An in-depth look at different topologies and test beds for networking
    - GENI, DeterLab
- How to read and appreciate the contents of research papers (3 pass method, etc.)
- Teamwork!
- How to make a poster, and in depth use of Powerpoint

Most important of all, a great appreciation for research and all the hard work that goes into producing it

References

[1] A. Juels and B. S. Kaliski, “PORs: Proofs of retrievability for large files,” In ACM CCS, pages 584-597, 2007.
[2] Amazon Inc., “Amazon customer agreement,” https://aws.amazon.com/agreement/, accessed on December 12, 2012.
[3] K. Bowers, A. Juels, and A. Oprea, “HAIL: a high-availability and integrity layer for cloud storage,” in Proc. of the 16th ACM conference on Computer and communications security, 2009.
[4] A. Juels and B. Kaliski Jr, “Pors: Proofs of retrievability for large files,” in Proceedings of the 14th ACM conference on Computer and communications security, 2007.
[5] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, “Provable data possession at untrusted stores,” in Proceedings of the 14th ACM conference on Computer and communications security, 2007.
[6] S. Kandula, D. Katabi, M. Jacob, and A. Berger, “Botz-4-sale: Surviv- ing organized ddos attacks that mimic flash crowds,” In Proc. NSDI (2005).
[7] A. Yaar, A. Perrig, and D. Song, “Fit: Fast internet traceback,” In Proc. IEEE Infocom (March 2005).
[8] M. Jensen, J. Schwenk, N. Gruschka, and L.L. Iacono, “On technical security issues in cloud computing,” Cloud Computing, 2009. CLOUD’09. IEEE International Conference on, 2009, pp. 109-116.
[9] J. Idziorek, M. Tannian, and D. Jacobson, “Detecting fraudulent use of cloud resources,” in Proc. 3rd ACM workshop on Cloud computing security workshop, New York, NY, USA, 2011, pp. 61-72.
[10] J. Idziorek and M. Tannian, “Exploiting cloud utility models for profit and ruin,” in Cloud Computing (CLOUD), 2011 IEEE International Conference on, 2011, pp. 33-40.
[11] Thomas E. Carroll, Michael Crouse, Errin W. Fulp and Kenneth S. Berenhaut, “Analysis of Network Address Shuffling as a Moving Target Defense”, Proc. of ICC, 2014.
[12] Huangxin Wang, Quan Jia, Dan Fleck, Walter Powell, Fei Li, Angelos Stavrou, “A moving target DDoS defense mechanism”, Elsevier Computer communications, 2014.
[13] Rui Zhuang, Su Zhang, Alex Bardas, Scott A. DeLoach, Xinming Ou, Anoop Singhal, “Investigating the Application of Moving Target Defenses to Network Security”, Proc. of ISRCS, 2013.
[14] Wei Peng, Feng Li, Chin-Tser Huang, and Xukai Zou, “A Moving-target Defense Strategy for Cloud-based Services with Heterogeneous and Dynamic Attack Surfaces”, Proc. of ICC, 2014.
[15] Quan Jia, Huangxin Wang, Dan Fleck, Fei Li, Angelos Stavrou, Walter Powell, “Catch Me if You Can: A Cloud-Enabled DDoS Defense”, Proc. of IEEE/IFIP DSN, 2014.

Thank you!

A special thanks to all the mentors and research directors of every project for their continued help and guidance for us undergraduates