Thimblerig: A Game-Theoretic, Adaptive, Risk-limiting Security System for Cloud Systems

Gautam Kumar, Brent Lagesse
University of Washington Bothell
Emails: {gautamk@uw.edu, lagesse@uw.edu}

***Abstract*—A significant portion of organizations and applications host client facing servers on cloud-based systems. As the first line of access into a system's services, these client-facing servers have a significant attack surface from network adversaries. Once compromised, these systems may be used to send spam, mine crypto, launch DDoS attacks, or used for other nefarious purposes. We propose an adaptive moving target defense that uses game theory to optimize the security and cost to the cloud system. This system leverages the fault-tolerant capabilities of cloud systems with large numbers of client facing servers and the virtualization of these client facing servers by strategically crashing random systems. As a result, an attacker who has compromised a system loses access to it and incurs the cost of having to re-compromise the system once they notice it has been lost. This approach drastically limits the amount of time that an attacker can utilize compromised systems and raises the overall investment required for that time. We have demonstrated via simulation a 90% reduction in the amount of time that an attacker has control over a compromised system for realistic scenarios based on previous data collection of live systems. This approach is agnostic to the method of compromise, so it is even effective against zero-day attacks.**

I. Introduction

Cloud services have become a critical part of both enterprise computing. To safeguard cloud deployments organizations are turning towards multi-layered security solutions [1], [2] which are common in physical security scenarios. One potential layer of security is a Moving Target Defense (MTD) which are designed to increase costs to attackers and reduce their probability of successful attack.

Much of the MTD research is done at the network level [3], [4], [5]. Network based MTD is relevant to cloud security, especially with the advent of SDNs being deployed in cloud infrastructure. We propose a game-theoretic MTD architecture for network security in the cloud, with the goal of significantly increasing the cost for an attacker to maintain and utilize compromised VMs as they infiltrate a virtual network. Our proposed architecture, Theimblerig[1], introduces an MTD in the form of ephemeral servers with a limited time to live (TTL). The TTL of each server is based on factors which are difficult to predict by an attacker. We provide evidence of effectiveness through simulation of our architecture to quantify the potential benefits that a defender may gain along with the increase in costs to an attacker.

Our system operates on the assumption that the cloud system it is deployed on expects disruption in the forms of VMs failing [6]. We purposely introduce failures to the system at random intervals and replaces them with fresh VMs. This has two primary effects that are the focus of this study. The positive effect is that it eliminates the foothold that an attacker has into a compromised system if a VM that they have compromised is crashed. The negative effect is that there is a resource cost to the system. This work describes an extension of our previous work where we demonstrated that the performance overheads of such a system [7]. In the remainder of this paper we explore the impact of these two effects through a game-theoretic framework that is used to develop an adaptive defense and argue that there exist realistic scenarios in which we improve network security while maintaining expected QoS. This approach is agnostic to the attacker's method of compromise, so it is effective even against zero-day attacks. We demonstrate that under realistic conditions, our system can reduce an adversary's expected attack rate by over 90%.

The contributions of our work are as follows:

- A game theoretic analysis and adaptation of the architecture when facing strategic attackers
- A simulation of our system using realistic parameters derived from extensive studies in security literature

II. Related work

In this section we provide a brief overview of the current research that has some relevance to our system.

A. Chaos Monkey

Chaos monkey [6] is a tool developed by Netflix. Chaos monkey tested reliability by randomly selecting virtual machine instances within Netflix's production infrastructure pool and terminated them. Terminating a virtual machine instance is equivalent to shutting down a computer and trashing its components. This means that the terminated instances are not recoverable. The primary reason for building such a destructive tool was to encourage engineers to design and build software services which are resilient unpredictable failure. Our proposed architecture aims to leverage the unpredictable nature of failure events in a controlled manner and apply that principle to securing cloud infrastructure using moving target defense.

[1] Code to reproduce simulations is available at https://github.com/SecurityInEmergingEnvironments/thimblerig-simulations

B. Current research in MTD

Much research has been done in moving target defenses[8], [9], [10]. Much of the primary focus in MTD research has been on implementing MTD at various levels in a network. For example Dunlop [3] developed MTD6, a system for leveraging the vast address space of IPv6 to improve user privacy and protect against targeted network attacks.

C. Game theory and Security

As cyber-security research has gained much more prominence over the last decade, many researchers have examined cyber security from a game theoretic perspective [11], [12], [13]. An example of such research is the work of Fan [14]. The authors model the interactions between defenders and attackers as a Stochastic game. A stochastic game is with multiple stages and one or more players. Each stage transition has a probability.

Similarly, Furuncu & Sogukpinar [15] analyze the security in IaaS cloud deployments using a normal form game to evaluate the costs and benefit that attackers and defenders encounter. Based on their analysis the authors claim that if an attacker attacks more than 76% of systems within an organization, then the cost of taking security measures would out weigh the benefits. Our work in section III-D2 is inspired by the work done on game theory based security mechanism for mobile P2P systems [16], [17]. The authors propose a game theoretic model for determining the pay-offs for an attacker and defender. Using these pay-offs the authors determine the mixed strategy Nash Equilibrium and the probability of attack.

The concept of using ephemeral servers as a mechanism for moving target defense is referenced by Dijk [18]. The authors propose a two player game where each player competes to maximize the amount of time that they have access to a resource while minimizing their costs.

III. Design

A. System Model

In general, we assume that system utilizes fault tolerance techniques for failed VMs, but specifically, we need the following properties for our system to work:

1) The VMs controlled by our system are completely interchangeable with freshly initialized VMs (significant state is not kept on the VMs)
2) The cloud system must have sufficient resources that the rate of forced failure from our system does not prevent it from meeting QoS requirements.

We demonstrate in [7] that the second assumption is not difficult to meet.

B. Attacker Model

We assume that the attacker's goal is to compromise our client facing machines via the network and then use them for as much time as possible to launch other attacks (e.g., DDoS), perform some task (e.g., password cracking), or to further compromise additional machines in our network. We make no assumptions about the techniques that an attacker has to compromise a system. In particular, **we specifically allow for the use of zero-day attacks**. We assume that identifying a victim system and launching the attack has a cost to the attacker (e.g., it takes some finite amount of time). Therefore, the attacker's goal is to maximize the cumulative system time that they have compromised machines while minimizing their own costs.

C. Architecture

Our architecture consists of two types of servers, Client Facing Servers (CFS) and a Central Trusted Authority (CTA). The CFS and CTA are classifications which refer to two commonly used server types. A CFS is any server which is capable of communicating outside of a virtual cloud network, while a CTA is any server which is inherently responsible for securing a sensitive resource such as an asset or service which needs to be protected from unauthorized access. Examples of sensitive resources include data stores such as databases, and authentication information to external services such as API Keys. Our work focuses on the CFSs as they are Internet-facing and most likely to be attacked.

1) Central Trusted Authority: The Central Trusted Authority consists of three primary components, A hash chain verifier, a storage back-end and a request proxy. The CTA performs three roles within the system, which are

- Create new hash chains
- Verify hash chains
- Proxy requests to resources

Creating a new hash chain: Hash chains are created by iteratively hashing a secret token T, n number of times. After the hash chain is created the CTA stores $H^n(T)$ in the storage backend and returns T and n to the client facing server. The secret token T is not stored by the CTA. The client facing server can now use the the secret token n number of times. Hash chains are used to authenticate client facing server requests which require access to a sensitive resource as described by [19]. The detailed analysis for the limiting the lifespan of a CFS through hashchains is described in [7]

D. Game definition

Our game consists of two players. The defender, who would be the security team for an organization, and any number of malicious actors who are working towards compromising the client-facing servers of the organization.

The malicious actor has two primary strategies, to attack by trying to compromise the system, or refrain from attacking. The malicious actor gains a pay-off only when they successfully compromise a system. Attacking a system also has costs associated with it. These costs include bandwidth usage, Command and Control server costs and Cost of being discovered.

1) Solving for Nash Equilibrium: To solve for the mixed strategy Nash equilibrium we defined a payoff matrix as shown in fig. 5. We then proceeded by setting the expected payoff for each action that a player could take, equal to the

TABLE I: Frequently used notation and simulation values

Notation	Value	
B_{att}	0.0005	Benefit of an attack
B_{use}	0.5	Benefit of utilization
C_{att}	3.256	Cost of attacking
C_{recon}	1	Cost of reconnaissance
C_{reset}	0.047	Cost of resetting
C_{vic}	5	Cost of being a victim
ET	$TTL - AT$	Exploitable time
MET	$MAXT - AT$	Max Exploitable Time
$MAXT$	$434 * 24$	Max Time
TTL	50	Server lifetime
AT	24	Attack time
P_{att}	see eq. (1)	Probability of attack
P_{reset}	see eq. (2)	Probability of reset

payoff for the alternative action. Solving these equations we obtained the probability of attack, defined by eq. (1), and probability of reset defined by eq. (2). Probability of attack determines whether an attacker will attack upon discovering a vulnerability or will wait until a later time.

$$P_{att} = \frac{C_{reset} + (B_{use} * MAXT) - (B_{use} * TTL)}{(C_{vic} * MAXT) - (C_{vic} * ET)} \qquad (1)$$

$$P_{reset} = \frac{C_{att} - B_{att} * MET}{B_{att} * ET - B_{att} * MET} \qquad (2)$$

2) Adapting to a Nash Equilibrium attacker: We examine two adaptive algorithms. These adaptive algorithms attempt to find the Nash equilibrium, and if the attacker is not using a Nash equilibrium strategy, they adapt to optimize against a non-rational attacker. The initial adaptive algorithm monitors (not shown due to space constraints) the attack rate and reduces the effective TTL of a CFS when the attack rate increases. We then modified the initial algorithm to better suit a Nash Equilibrium attacker by factoring in utility change rather than attack rate. This change offers the algorithm the ability to adapt to multiple factors beyond merely the attack rate. The adaptive approach updates the TTL as a function of how many attacks are detected during the process shown in figure fig. 1. Utility for the defender is computed as shown in eq. (3) where T_{exp} represents the amount of time that an attacker is able to exploit a compromised device.

$$U_{def} = (TTL * B_{use}) - C_{res} - (T_{exp} * C_{vic}) \qquad (3)$$

We use selective forensic scanning to estimate the actions of the adversary and adapt the CFS lifetime. Scanning is considered to be an expensive operation. This implies that a scan cannot be performed on all CFS instances. A technique similar to packet sampling [20] can be used to sample CFS instances for analysis. Such a sampling technique merely implies that attacks are detected with a probability of $P(D)$. This scanning strategy is highlighted in figure 1.

IV. Results

1) Background: In this section we present the results of a data-driven simulated study. We demonstrate evidence that our system provides significant improvements in security. We show both an exploration of the parameter space for the system and the results of an adaptive, game-theoretic solution. Parameters for the simulation were draw from extensive previous work on the statistical behavior of malicious actors [21]. These parameters can be found in table I.

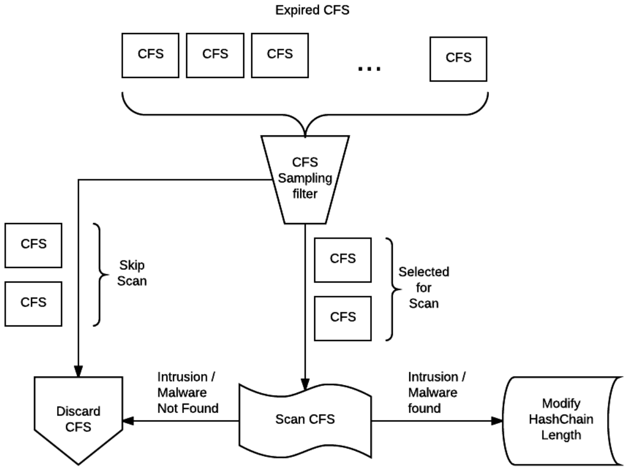

Fig. 1: CFS scan strategy based on sampling

A. Effects of TTL on successful attacks

This simulation illustrates the effects of an adaptive algorithm against a naïve attacker who attacks at a constant rate. The adaptive algorithm monitors the attack rate and reduces the effective TTL of a CFS when the attack rate increases. An attack is successful in this simulation when the attacker has enough time to perform reconnaissance, exploit a vulnerability, and have a pre-defined minimum amount of time left over in the CFS's TTL to effectively utilize the server's resources. If any of these criteria are not met the attack is considered a failure.

Figure 3 demonstrates the effect of adaptive TTL calculation. The X-axis is the measurements taken at various TTLs starting points and the Y-axis depicts a count of attacks.

Each colored line on fig. 3 represent different types of measurement. The blue line represents a count of successful attacks, while the orange line depicts the number of successful attacks which were discovered using the scanning strategy described in section section III-D, and the green line presents the number of failed attacks. The simulation ran for $10,000$ units of simulation time for each point on the X-axis. Initial TTL values ranging from 1 to $10,000$ were simulated in increments of 100.

By comparing figures 2 and 3 we can conclude that our naïve adaptive algorithm increases the effectiveness of a limited lifetime server in reducing the number of successful attacks, especially with lower values of TTL. This simulation inspired us to simulate a Nash Equilibrium attacker (see

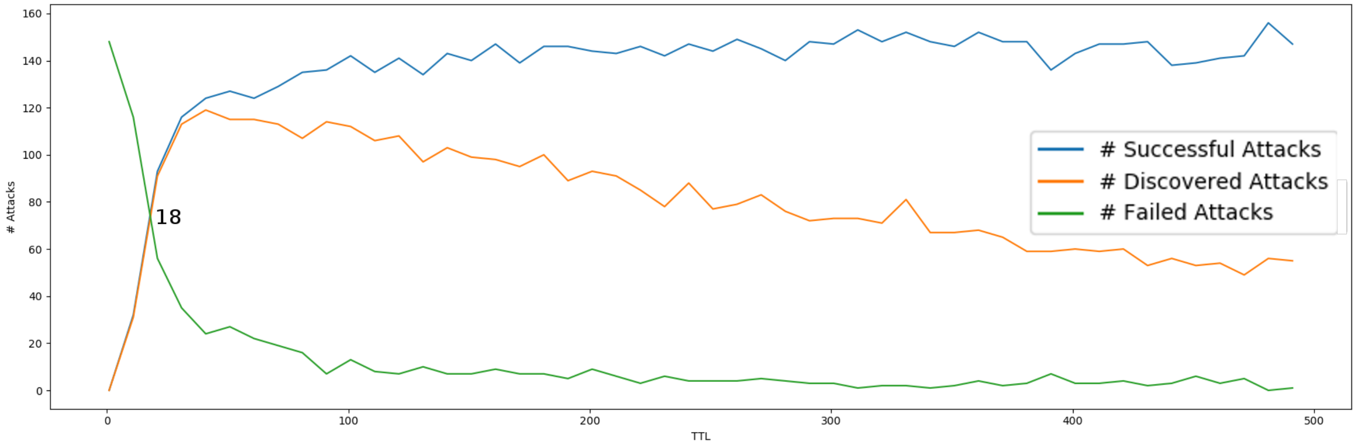

Fig. 2: Non-Adaptive TTL System

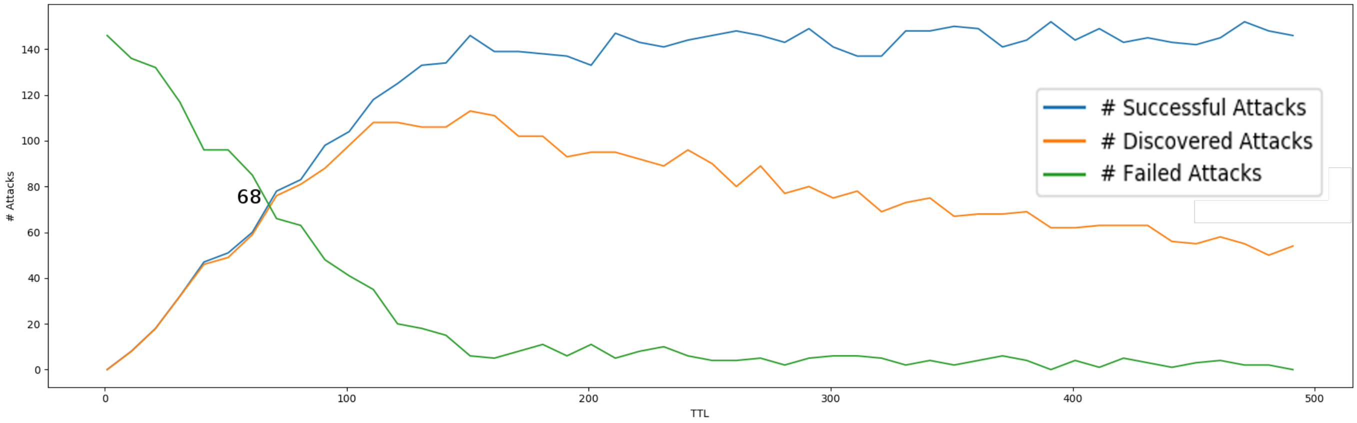

Fig. 3: Adaptive TTL System

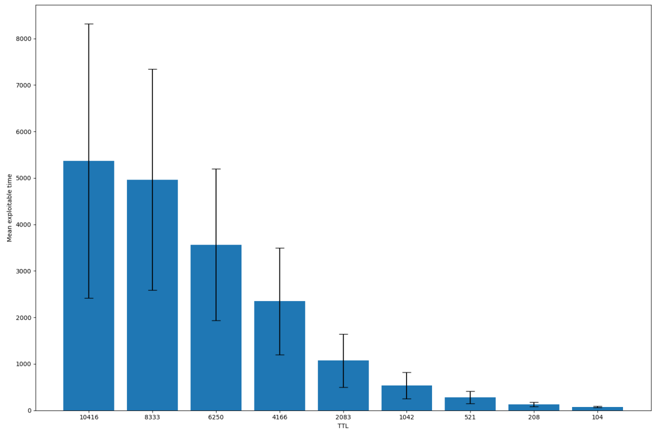

Fig. 4: Mean exploitable time and its standard deviation (Lower is better)

section III-D2) and modify our naïve adaptive algorithm to be based on the defender's utility rather than the rate of attack.

B. *Exploitable time*

Reducing successful attacks is not the only positive effect of our system. Exploitable time is the amount of time that an attacker has access to a cloud system after the system has been successfully compromised. Measuring exploitable time enables us to understand the effect of our system on the attacker's goals. Value for an attacker could be quantified using many different parameters. For example, an attacker could sell server time to botnets. The value an attacker can derive from a system is in direct relation to the amount of exploitable time that is available. The simulation setup for this simulation is equivalent to the setup described in table I.

Figure 4 visualizes the mean exploitable time for each value of TTL. The Y axis represents the exploitable time and each bar describes the mean exploitable time for its corresponding value of TTL. Lower exploitable time is better as the attacker has less time to access the system. As we can see the mean exploitable time decreases significantly as TTL decreases.

		Attacker			
		Attack		No Attack	
Defender	Reset	$B_{att} * ET - C_{att} - C_{recon}$	-4.24	$-C_{recon}$	-1
		$B_{use} * TTL - C_{vic} - C_{reset}$	-105.05	$B_{use} * TTL - C_{reset}$	-24.95
	No Reset	$B_{att} * MET - C_{att} - C_{recon}$	0.94	$-C_{recon}$	-1
		$B_{use} * MAXT - C_{vic}$	-46872	$B_{use} * MAXT$	5208

Fig. 5: Payoff Matrix with formulations for each payoff with example values from [21]

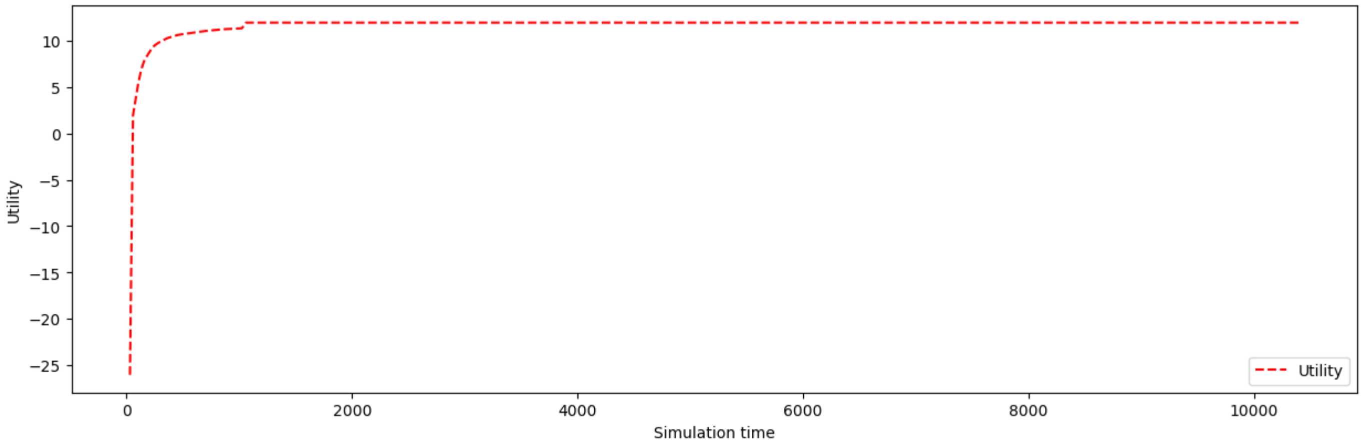

Fig. 6: Results of Adaptive algorithm against Nash Equilibrium Attacker

C. Game Theoretic Adaptation

1) Analysis: Upon solving eq. (1) using the values specified in table I we get a value 9.9% for P_{att}. This means that it is in the attacker's best interest to reduce the amount of attacks they launched on our system by over 90%.

Figure 6 illustrates the effects of applying the adaptive algorithm. In each figure, the X-Axis represents simulation time and the Y-Axis represents utility. The algorithm initially adjusts the TTL of the CFSs until it converges to approximate the Nash equilibrium value and maintains a steady state utility. These results are the average of 10,000 simulations.

V. Discussion of Limitations

There are several limiting factors that we must consider in our analysis. We simplify an attackers ability to exploit a system using a probability of attack (P_{att}) and a fixed span of attack time. There may be zero day vulnerabilities, for example, that break our assumptions. If an attacker exploits zero-day vulnerabilities the attacker may be able to gain near instantaneous access to a CFS; however, the attacker would still be limited by the TTL. We can increase the reconnaissance cost to the attacker by leveraging artificial diversity in the VMs [22], [23], so that it is less likely that an attacker will always successfully identify a vulnerable CFS.

Cloud infrastructure may suffer from unpredictable cloud latencies which we do not model in our simulation. A possible solution for handling start up latency is to over-provision server infrastructure or utilize a server pool[7]. Research into reliability engineering and the usage of spot instances [24], [25] also offers multiple solutions to the problem of reliability with cloud VMs with limited lifetime.

In our simulation on the effects of TTL on the number of successful attacks (see section IV-A) we utilized a probability of attack (9.9%) computed using the the values of TTFC and TBC from the large scale study[21]. This value may not represent every attack scenario as the study was conducted on windows desktop computers in a large organization, this is unlike many cloud infrastructure deployments that are Linux based [26] and may their attackers may have different statistical properties.

VI. Conclusions

We proposed a cloud architecture which provides a layer of Moving Target Defense through limiting the lifetime of Client-Facing Servers. By limiting the lifetime of servers we also reduce the amount of time that an attacker has access to a server. We tie the lifetime of server to factors which are difficult to predict by an attacker such as request rate and utility. By tying the lifetime of servers to difficult to predict factors, we increased the uncertainty and apparent complexity of our system to achieve Moving Target Defense.

We used simulations to perform a case study evaluation of our system. Our simulations show that our architecture vastly reduces the cost of being exploited by reducing an attacker's access to total exploitable time. Finally we modeled our system as a game to solve for its mixed strategy Nash Equilibrium. We demonstrated that with realistic data our system is able to cause a rational attacker to reduce their attack rate by over 90% in order to perform their attacks most effectively.

In the future we intend to pursue ethical methods for evaluating our system in the wild without enabling attackers to utilize our VMs for malicious purposes. We also intend to extend a game-theoretic moving target defense to the Central Trusted Authority so that we will have greater trust that it has not been compromised.

Acknowledgment

This work is partially funded by the National Science Foundation, Grant No. 1853953.

References

[1] M. Yildiz, J. Abawajy, T. Ercan, and A. Bernoth, "A Layered Security Approach for Cloud Computing Infrastructure," in *2009 10th International Symposium on Pervasive Systems, Algorithms, and Networks*, Dec. 2009, pp. 763–767.

[2] A. Panwar, R. Patidar, and V. Koshta, "Layered security approach in cloud," in *3rd International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom 2011)*, Nov. 2011, pp. 214–218.

[3] M. Dunlop, S. Groat, W. Urbanski, R. Marchany, and J. Tront, "MT6d: A Moving Target IPv6 Defense," in *2011 - MILCOM 2011 Military Communications Conference*, Nov. 2011, pp. 1321–1326.

[4] D. C. MacFarland and C. A. Shue, "The SDN Shuffle: Creating a Moving-Target Defense Using Host-based Software-Defined Networking," in *Proceedings of the Second ACM Workshop on Moving Target Defense*, ser. MTD '15. New York, NY, USA: ACM, 2015, pp. 37–41. [Online]. Available: http://doi.acm.org/10.1145/2808475.2808485

[5] P. Kampanakis, H. Perros, and T. Beyene, "SDN-based solutions for Moving Target Defense network protection," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium on a*, Jun. 2014, pp. 1–6.

[6] A. Basiri, N. Behnam, R. d. Rooij, L. Hochstein, L. Kosewski, J. Reynolds, and C. Rosenthal, "Chaos Engineering," *IEEE Software*, vol. 33, no. 3, pp. 35–41, May 2016.

[7] G. Kumar and B. Lagesse, "Limited Use Cryptographic Tokens in Securing Ephemeral Cloud Servers," in *Proceedings of the 3rd International Conference on Information Systems Security and Privacy*. Porto, Portugal: SCITEPRESS, Feb. 2017, p. 447. [Online]. Available: http://faculty.washington.edu/lagesse/publications/LimitedCryptTokens.pdf

[8] H. Alavizadeh, S. Aref, D. S. Kim, and J. Jang-Jaccard, "Evaluating the Security and Economic Effects of Moving Target Defense Techniques on the Cloud," *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 4, pp. 1772–1788, Oct. 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9733785/

[9] M. Torquato and M. Vieira, "Moving target defense in cloud computing: A systematic mapping study," *Computers & Security*, vol. 92, p. 101742, May 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404820300286

[10] T. Penner and M. Guirguis, "Combating the Bandits in the Cloud: A Moving Target Defense Approach," in *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, May 2017, pp. 411–420. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7973727

[11] M. H. Manshaei, Q. Zhu, T. Alpcan, T. Bacşar, and J.-P. Hubaux, "Game theory meets network security and privacy," *ACM Computing Surveys (CSUR)*, vol. 45, no. 3, p. 25, 2013. [Online]. Available: http://dl.acm.org/citation.cfm?id=2480742

[12] Q. Zhu and T. Başar, "Game-theoretic approach to feedback-driven multi-stage moving target defense," in *International Conference on Decision and Game Theory for Security*. Springer, 2013, pp. 246–263. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-319-02786-9_15

[13] C. A. Kamhoua, L. Kwiat, K. A. Kwiat, J. S. Park, M. Zhao, and M. Rodriguez, "Game Theoretic Modeling of Security and Interdependency in a Public Cloud," in *2014 IEEE 7th International Conference on Cloud Computing*, Jun. 2014, pp. 514–521.

[14] G. Fan, H. Yu, L. Chen, and D. Liu, "A Game Theoretic Method to Model and Evaluate Attack-Defense Strategy in Cloud Computing," in *2013 IEEE International Conference on Services Computing*, Jun. 2013, pp. 659–666.

[15] E. Furuncu and I. Sogukpinar, "Scalable risk assessment method for cloud computing using game theory (CCRAM)," *Computer Standards & Interfaces*, vol. 38, pp. 44–50, Feb. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0920548914000853

[16] B. Lagesse and M. Kumar, "A Novel Utility and Game-Theoretic Based Security Mechanism for Mobile P2p Systems," in *2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom)*, Mar. 2008, pp. 486–491.

[17] B. Lagesse, M. Kumar, and M. Wright, "AREX: An adaptive system for secure resource access in mobile P2p systems," in *Peer-to-Peer Computing, 2008. P2P'08. Eighth International Conference on*. IEEE, 2008, pp. 43–52. [Online]. Available: http://ieeexplore.ieee.org/abstract/document/4627257/

[18] M. Van Dijk, A. Juels, A. Oprea, and R. L. Rivest, "FlipIt: The game of "stealthy takeover"," *Journal of Cryptology*, vol. 26, no. 4, pp. 655–713, 2013. [Online]. Available: http://link.springer.com/article/10.1007/s00145-012-9134-5

[19] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981. [Online]. Available: http://dl.acm.org/citation.cfm?id=358797

[20] D. Brauckhoff, B. Tellenbach, A. Wagner, M. May, and A. Lakhina, "Impact of Packet Sampling on Anomaly Detection Metrics," in *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '06. New York, NY, USA: ACM, 2006, pp. 159–164. [Online]. Available: http://doi.acm.org/10.1145/1177080.1177101

[21] H. Holm, "A Large-Scale Study of the Time Required to Compromise a Computer System," *IEEE Transactions on Dependable and Secure Computing*, vol. 11, no. 1, pp. 2–15, Jan. 2014.

[22] B. Cox, D. Evans, A. Filipi, J. Rowanhill, W. Hu, J. Davidson, J. Knight, A. Nguyen-Tuong, and J. Hiser, "N-variant Systems: A Secretless Framework for Security Through Diversity," in *Proceedings of the 15th Conference on USENIX Security Symposium - Volume 15*, ser. USENIX-SS'06. Berkeley, CA, USA: USENIX Association, 2006, event-place: Vancouver, B.C., Canada. [Online]. Available: http://dl.acm.org/citation.cfm?id=1267336.1267344

[23] D. Williams, W. Hu, J. W. Davidson, J. D. Hiser, J. C. Knight, and A. Nguyen-Tuong, "Security through Diversity: Leveraging Virtual Machine Technology," *IEEE Security Privacy*, vol. 7, no. 1, pp. 26–33, Jan. 2009.

[24] S. Yi, D. Kondo, and A. Andrzejak, "Reducing Costs of Spot Instances via Checkpointing in the Amazon Elastic Compute Cloud," in *2010 IEEE 3rd International Conference on Cloud Computing*, Jul. 2010, pp. 236–243.

[25] N. S. V. Rao, S. W. Poole, F. He, J. Zhuang, C. Y. T. Ma, and D. K. Y. Yau, "Cloud computing infrastructure robustness: A game theory approach," in *2012 International Conference on Computing, Networking and Communications (ICNC)*, Jan. 2012, pp. 34–38.

[26] "OS/Linux Distributions using Apache." [Online]. Available: https://secure1.securityspace.com/s_survey/data/man.201904/apacheos.html