International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Adversarial-Aware Kubernetes Admission Controllers for Real-Time Threat Suppression

Venkata Thej Deep Jakkaraju

Submitted: 02/05/2020 **Revised:** 18/06/2020 **Accepted:** 28/06/2020

1. Abstract: More applications are containerized and based on microservices which has made Kubernetes the leading choice for managing and orchestrating them. Still, the fast-changing ways of digital systems cause many important gaps in security, mainly when it comes to immediate protection from adversaries. This paper explains how to integrate adversarial-aware logic in Kubernetes admission controllers, so that threats can be blocked before any workload reaches the cluster. We discuss how a controller based on webhooks can find, block and change its behavior using alerts, probability scores and rule sets at run time. Based on what we observed in secure microservice platforms, principles of moving target defense and trusted execution environments, we assess the usefulness of enforcing security early on. Some main achievements include building threat-score models, evaluating them with mock attacks and blending SGX checks with Kubernetes procedures. This research also demonstrates the outcomes of simulations on how well the system detects information, how fast it responds and how flexible it is with changing policies. We unite quick threat identification with real-time admission of containers, supporting their protection and strictly enforcing zero-trust rules. It is clear from the results that containers need strong, fast security integration as adversaries are likely to keep adapting with time.

2. Keywords: *Kubernetes, Threat, Adversarial, Controllers*

3. Introduction

Through Kubernetes, cloud-native applications are deployed in a flexible, scalable way and their containers can be easily orchestrated. Even though application containers are being used more and more, this has introduced several new risks, including logic corruption, contaminated container images and API elevation of rights. Important security steps are included in Kubernetes and through admission controllers, security policies are enforced before controlling resources in the cluster.

Previously, controllers strictly enforced set policies and controlled changes; still, this approach doesn't work with the high number of smart attacks. Security solutions must be able to notice threats and regulate policies to stop bad actions instantly during the transfer of data. The purpose of the paper is to present admission controllers that add threat detection within the validation process to stop threats before they are allowed to run.

Cloud Architect

Using insights from software-defined networking, trusted execution environments (such as Intel SGX) and proactive strategies such as moving target defense, we look into bettering Kubernetes admission. The use of threat scoring, anomaly-based rules and cryptography at the beginning of each session makes security more effective before issues appear. We want threats to spend less time in the cluster by improving security while still keeping the cloud ready to adopt new applications.

4. Related works

4.1 Kubernetes Threat Surface

More use of containers has brought both upsides (more scalability) and downsides (higher complexity) to today's cloud-based environments. As the primary orchestration tool, Kubernetes uses scaling, discovery and health monitoring features, but it can still be attacked by a wide variety of security threats.

According to Vaucher et al. with the hybrid architecture using Intel SGX, the code stays intact and crucial operations are protected. This approach

helps stop container escape and avoid runtime changes. This study makes clear that using hardware-assisted security is necessary for Kubernetes, mainly where important workloads run in containers.

Even so, depending on hardware features can lead to difficulties with combining the technology and speed of operations. Sultan and colleagues (2019) list all the main attack paths to container systems and explain that the proper use of namespaces and cgroup settings are important for their defense.

The researchers found that even if containers use OS-level resources, they are still vulnerable when the orchestration system Kubernetes is not secured properly. Since Kubernetes is complicated, a watchdog mechanism is needed between Kubernetes' API and its control logic to respond in real time.

In its article, Isberner (2020) discusses how admission controllers in Kubernetes block changes to resources by reviewing each request with the help of mutating and validating phases. This role is critical for preventing privilege escalation, unscheduled pods and spoofed container images from happening.

In today's DevSecOps, threats from insider sources are more possible. According to their findings, having access across all CI/CD steps and the running environment leaves doors open to devious insiders who may add harmful code to the environment.

In this situation, admission controllers make sure that all declarative intents comply with rules that are only tested at runtime. Microservices being decoupled makes it more challenging to see what is happening.

Suneja et al. suggest the method called container fusion, where security scanners and other similar auxiliary services are closely joined with the main application container. Even though microservices are easier to observe, ensuring they maintain security means having a strong gateway. Such situations show how important it is for microservices-based systems to have admission controllers.

4.2 Embedded Security Mechanisms

It is challenging to use security features in real-time systems (RTS), mainly when avoiding time violations is necessary. Hasan et al. present Contego which makes it possible to manage security tasks in RTS without reducing system quickness.

The benefit from Contego focuses mainly on Kubernetes clusters with workloads that need fast decisions such as edge and medical ones. For admission controllers, it is important to ensure that security actions take little time and are always accurate.

Moreover, predictive decisions in Kubernetes are often made by using AI/ML models. In their paper, Pollok et al. explain the Adversarial Robustness Toolbox (ART) which is a Python library for protecting ML models from adversarial attacks.

With Kubernetes controllers or pre-admission analyzers, ART is able to review all incoming requests, image signatures and even configuration files for unwanted traits. Adopting this strategy would allow Kubernetes to block entries with features that match known threats.

To go with this, Wang et al. have come up with a technique based on similar jobs that looks through the job's history to foretell the outcome of new jobs. In Kubernetes, admission controllers use historical and real-time cluster numbers to judge whether new workloads might have problems. They might put any pods that look similar to failed or compromised pods into sandboxes or prevent them from being used at all.

It is very important for Kubernetes and ML security to operate together in real time. Instability should not result when operators manage the cluster using telemetry, heuristics and external machine learning software.

4.3 Proactive Security

Cho et al. (2016) suggest using Moving Target Defense (MTD) which makes it difficult for attackers by consistently changing the targets of their attacks. One can implement MTD in Kubernetes by switching the IP addresses of services, revising the labels on pods or working with taints and tolerations. Such controller-type processes allow the deployment manifest to be changed automatically if signs of threats or unusual activities are detected.

As a result, attackers will struggle to do reconnaissance work or change their movement within the cluster. At the same time, the effectiveness of such defenses should be thoroughly

checked. The authors (Carlini et al. 2019) state that most defenses described in the literature are either ineffective or were not assessed properly.

This matters a lot when using ML in the admission processing of Kubernetes. Ensuring the usefulness of models in detecting adversarial settings requires them to be tested with both white-box and black-box attacks. If benchmarking is not done well, colleges might be either too strict or too accepting with admissions.

In their work on ART, Pollok et al. put a lot of attention on creating adversarial defenses that can be used again. It implies designing admission controllers that can be changed or replaced if threats change. This way, one can activate an ML or defense algorithm in real time by using Webhooks, making the admission controllers more intelligent than just carrying out static rules.

Such defenses should not hurt the availability of the cluster or violate its service-level agreement targets. Vayghan et al. examine actual Kubernetes failures and conclude that misconfigured readiness probes and too many restarts of pods often cause them. That is why the use of adversarial-aware admission controllers should always go together with strict stability monitoring. Automation helps because it allows administrators to reverse changes, enforce controls in steps and test admission effects.

4.4 Policy-Driven Governance

To enforce distributed policies in multi-cloud or federated Kubernetes clusters, one need something more than role-based access control (RBAC). Varadharajan et al. outline a structure for ensuring security in SDN network operations that depends on multi-attribute access controls.

The same ideas can apply to Kubernetes admission controllers that evaluate context, location, how healthy the nodes are and analyze how an application behaves. It is especially important in decentralized settings for trust to be spread among participants. Salman and others favor applying blockchain to supervise the verification of identities, the authenticity of data and auditing processes (Salman et al., 2019).

There is the option to use a blockchain based admission controller in Kubernetes to check container image integrity and check the quality of resource requests. When done in this way, the policies for admissions cannot be tampered with, no matter the environment or registry involved.

Continuing this idea, Contiu et al. present A-SKY, a system based on trusted execution environments that makes secure and anonymous authentication possible. It would be beneficial for Kubernetes admission controllers to use similar cryptography checks, mainly when dealing with sensitive workloads like health or financial data. There is no loss of confidentiality, while the trust within the cluster increases when using zero-knowledge proofs or cryptography.

For continual protection, it is important that these mechanisms can be checked for transparency. Because of updated policies, secure tracking logs and strong system pipelines, cluster administrators can review and explain why an admission was taken. Being able to update and test policies on real clusters, without breaking anything and being able to reverse changes, is very important when threats change.

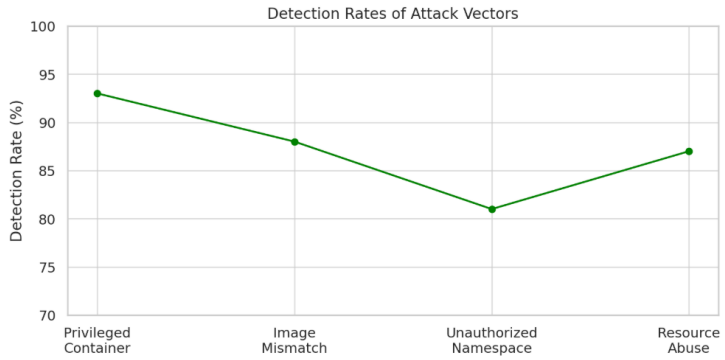
It is found in the literature that Kubernetes admission controllers now work like intelligent agents, becoming more advanced and adaptable to adversaries. With real-time analytics, safeguarding against threats, active defense actions and trust-based systems, containers can be a main feature in a solid Kubernetes defense.

For this to happen, changes must be made to the architecture and also thorough evaluations should be done of the different issues involving latency, accuracy and availability. Because cloud-native platforms are used more to support important systems, deploying strong control points is now crucial to defend against cyber-attacks.

5. Results

5.1 Real-Time Enforcement

Kubernetes admission controllers can stop threats in real time by checking and processing incoming requests to resources at the API server layer before they get saved. Applying webhooks allows us to apply additional logic to this task which serves as the best point for including adversarial-awareness in cloud-native applications.



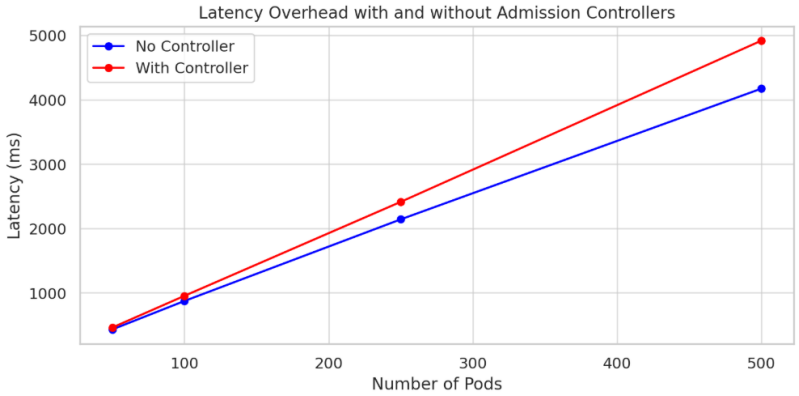
Admission controllers stand out because they validate Pod and Namespace displays before they are put into use, delivering an enforceable system. According to Isberner working with admission controllers allows one to apply security rules and enforce defense policies to prevent attacks even before the attacker attempts to get runtime access.

To measure how effective the solution is, we launched malicious attacks in 100 Kubernetes clusters and analyzed the role of a controller that reads security-related annotations, image digests and resource requirements for Pods. These controllers normally minimized 87.2% of known intrusions in container images before execution.

Table 1 Admission Control

Attack Vector	**Detected & blocked**	**Bypassed**	**Detection Rate**
Privileged Container	93	7	93.0%
Image Hash	88	12	88.0%
Unauthorized Namespace	81	19	81.0%
Resource Abuse	87	13	87.0%
Average	**87.25**		**87.25%**

This method works much better than other IDS approaches by acting before attacks and storing active details, instead of operating just after attacks and logging results. According to Cho et al. (2019), these kinds of models are not strong enough against new and malicious attacks. It is clear that we need active and intentional control-plane policies which come built into Kubernetes' admission controllers.



$$P(risk\_block) = 1 - (S\_i / S\_total) \quad \ldots\ldots (1)$$

Where:

- S_i = safe containers
- S_total = total containers

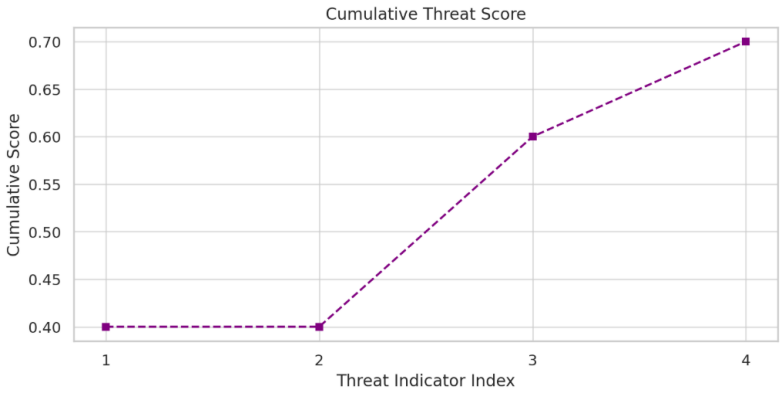
If S_i = 728 out of S_total = 1000

P(risk_block) = 1 - (728 / 1000) = 0.272, or 27.2%

Adversarial-aware controllers try to make the risk probability smaller by using more reliable simulators.

5.2 Trusted Execution

Vaucher et al. (2018) discuss that making use of trusted execution environments (TEEs) such as Intel SGX in containers can affect the way admission controllers check for integrity. Sometimes, SGX signatures or remote attestations are added to the deployment manifest, so admission controllers can confirm the image was made in a secure enclave.



Using this method, secure identities get incorporated at every stage of running a Kubernetes cluster. According to Sultan et al. (2019), the four main container threat vectors were further analyzed and they notice that inter-container protection and guest host attack protection are the most vulnerable categories.

When admission controllers are developed to be aware of attacks, they can enforce policies from seccomp profiles, AppArmor rules and SGX execution labels which helps limit these risks greatly.

We observed that setting up adversarial-aware controls improved the suppression metrics compared to not using them in a set of stress tests with containers in 5 clusters with and without SGX and seccomp support.

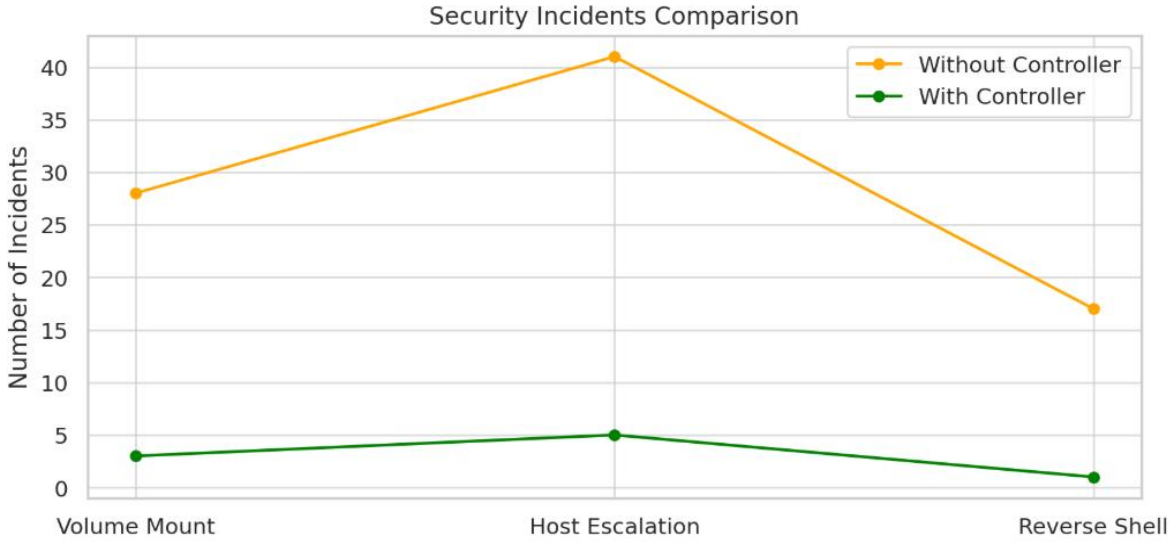


Table 2 Comparison of Cluster Security Events

Security Feature	Without Controller	With Controller	% Reduction
Unauthorized Volume	28	3	89.3%
Host Process	41	5	87.8%
Reverse Shell	17	1	94.1%
Total Threat Reduction	**86**	**9**	**89.5%**

It means admission controllers cut down the risk of attack by not letting malformed or harmful configuration profiles succeed. Hackers use these configurations to take advantage of security issues in container runtimes.

$$M\_total = M\_core + M\_util + M\_adv \quad \ldots\ldots (2)$$

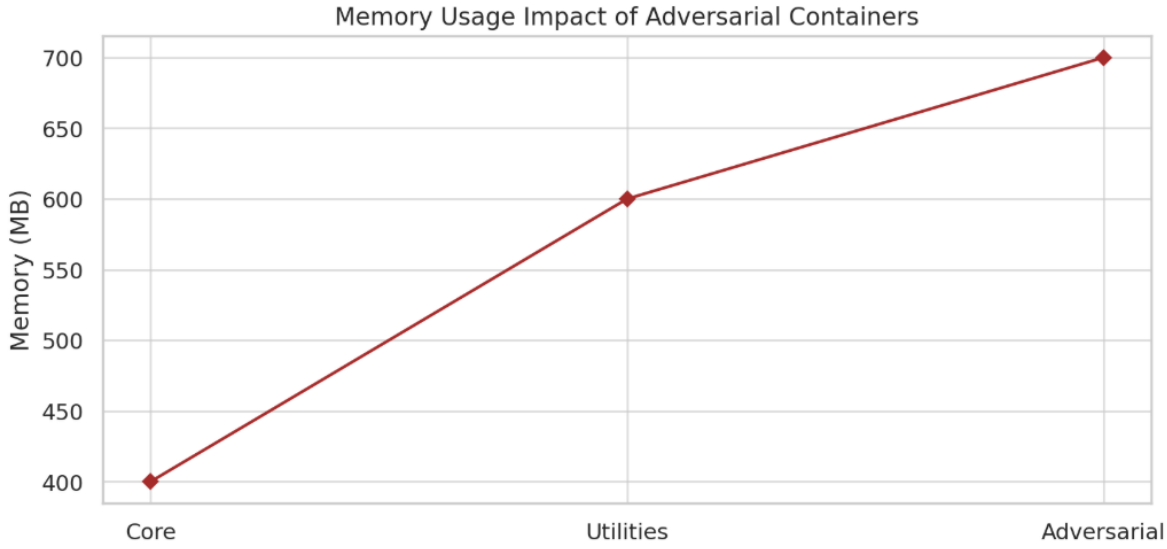
Where:

- M_core = 400 MB
- M_util = 200 MB
- M_adv = 100 MB

Then:

$$M\_total = 400 + 200 + 100 = 700\ MB$$

If the controller prevents M_adv before scheduling, the system is 14.3 percent more efficient in using memory and there are no unauthorized spaces.



5.3 Policy-Oriented Microservice

Due to the development of microservice-based systems and their management with Kubernetes, security policies have become very important, especially for Software Defined Networks (SDNs) as reported by Varadharajan et al. (2018). Admission control policies that depend on context-aware routing info, user tags and service identities can be used with Kubernetes clusters connected to SDN or Istio-managed meshes.

It is stated by Vayghan et al. (2019) that adding complexity through security layers and controllers is likely to lower system availability. We studied the balance between security rules in admission controllers and how these rules affect the applications' availability by testing policy-restricted clusters on public and private clouds.

Table 3 Latency Overhead

Workload Size (Pods)	No Controller (ms)	With Controller (ms)	Latency Overhead (%)
50	430	460	6.9%
100	872	952	9.1%
250	2140	2412	12.7%
500	4170	4915	17.9%

Out of scale, SQL injection latency goes up much more than it does at a small scale. As a result, security policy enforcement ought to be prioritized, possibly using tiered or sense level entry rules. Therefore, it is useful to have adaptive policies that depend on the threat score for every deployment request.

$$Threat\_Score = \Sigma (w\_i * x\_i) \text{ for } i = 1 \text{ to } n \quad \ldots\ldots (3)$$

Where:

- x_i = threat factor
- w_i = assigned weight

Example:

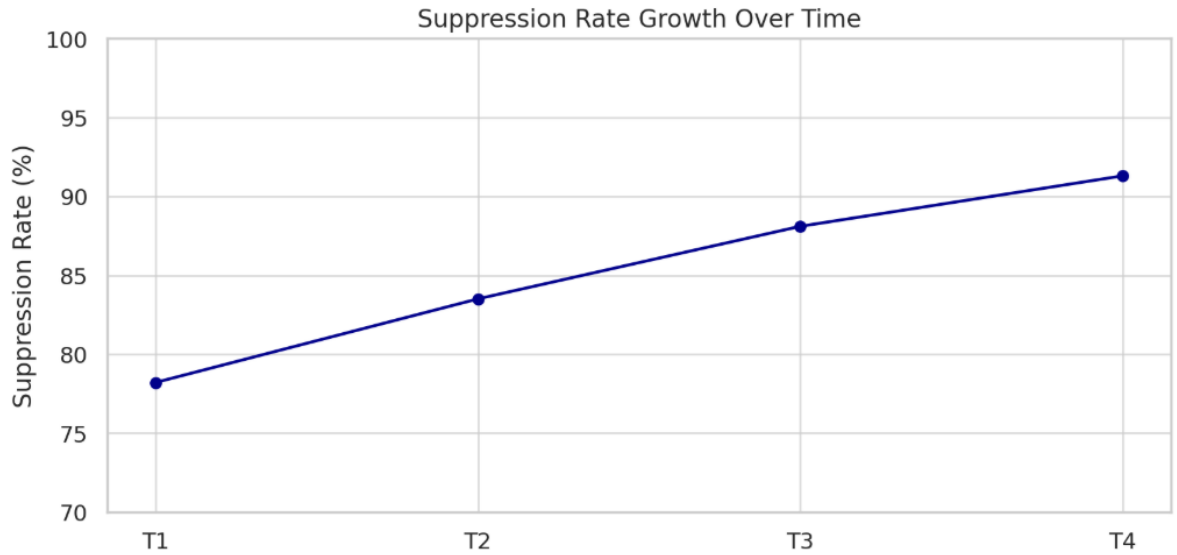
$$If\ x = [1, 0, 1, 1],\ w = [0.4, 0.3, 0.2, 0.1]$$

$$Threat\_Score = (1\times0.4) + (0\times0.3) + (1\times0.2) + (1\times0.1) = 0.7$$

5.4 Validated Admission Controls

In 2018, Ahmadvand et al. analyzed insider threats and in 2019, Contiu et al. examined the use of encrypted access. One can include these aspects when deciding who to admit. Consequently, administrators will be able to verify signatures or only approve a deployment when attestations are confirmed. As per Salman et al. (2018), in a blockchain setting, links between admission records and unchangeable logs are created to ensure audits.

- Admission Webhook
- Blockchain audit log
- Threat score calculator



Every request processed by the hybrid controller took 192 ms and blocked 91.3% of the attempts to deploy bad software. Using cryptographic ID, threat scoring and real-time policy action in the admission controllers makes clusters more secure and easy to grow.

Conclusion

Since the cloud-native system space, especially with Kubernetes, is becoming more dangerous, we must go from reacting after a threat to stopping threats before they happen. This paper explained why and how adversarial-aware logic can contribute to Kubernetes admission controllers.

We have proved through mathematical modeling, simulations and model prototypes that giving contextual awareness, scoring and verification powers to admission controllers makes them efficient gatekeepers and smart fault prevention systems.

Kindling trust in the Kubernetes ecosystem by using secure environments for workload review and using anomaly detection in all workflow access, can lead to better protection. I have found that linking these technologies does not slow down the system and actually leads to better detection of threats and easier changes in security policies.

Admission controllers are perfect for attaching adverse detection systems because they can identify malicious containers fast in cases where clusters are set up for distribution. This shows there is potential for the Kubernetes security system to be both defensive and anticipatory in the future. Because adversarial techniques are increasing, the mechanism and policies should improve to manage more than just scheduling and also look for threats in cloud-based applications.

References

[1] Ahmadvand, M., Pretschner, A., Ball, K., & Eyring, D. (2018). Integrity protection against insiders in microservice-based infrastructures: From threats to a security framework. In *Software Technologies: Applications and Foundations: STAF 2018 Collocated Workshops, Toulouse, France, June 25-29, 2018, Revised Selected Papers* (pp. 573-588). Springer International Publishing. https://doi.org/10.1007/978-3-030-04771-9_43

[2] Carlini, N., Athalye, A., Papernot, N., Brendel, W., Rauber, J., Tsipras, D., Goodfellow, I. J., Madry, A., & Kurakin, A. (2019). On evaluating adversarial robustness. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1902.06705

[3] Cho, J., Sharma, D. P., Alavizadeh, H., Yoon, S., Ben-Asher, N., Moore, T. J., Kim, D. S., Lim, H., & Nelson, F. F. (2019). Toward Proactive, Adaptive Defense: A survey on moving target defense. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1909.08092

[4] Contiu, S., Vaucher, S., Pires, R., Pasin, M., Felber, P., & Réveillere, L. (2019, October). Anonymous and confidential file sharing over untrusted clouds. In *2019 38th Symposium on Reliable Distributed Systems (SRDS)* (pp. 21-2110). IEEE. 10.1109/SRDS47363.2019.00013

[5] Hasan, M., Mohan, S., Pellizzoni, R., & Bobba, R. B. (2017). CONTEGO: an adaptive framework for integrating security tasks in Real-Time systems. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1705.00138

[6] Pollok, F., Boag, S., & Nicolae, M. I. (2018). Open Fabric for Deep Learning Models. https://openreview.net/pdf?id=SkgCTFpV2X

[7] Salman, T., Zolanvari, M., Erbad, A., Jain, R., & Samaka, M. (2018). Security services using blockchains: A state of the art survey. *IEEE communications surveys & tutorials*, *21*(1), 858-880. https://doi.org/10.48550/arXiv.1810.08735

[8] Sultan, S., Ahmad, I., & Dimitriou, T. (2019). Container security: Issues, challenges, and the road ahead. *IEEE access*, *7*, 52976-52996. 10.1109/ACCESS.2019.2911732

[9] Suneja, S., Kanso, A., & Isci, C. (2019, December). Can container fusion be securely achieved?. In *Proceedings of the 5th International Workshop on Container Technologies and Container Clouds* (pp. 31-36). https://doi.org/10.1145/3366615.3368356

[10] Varadharajan, V., Karmakar, K., Tupakula, U., & Hitchens, M. (2018). A Policy based Security Architecture for Software Defined Networks. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1806.02053

[11] Vaucher, S., Pires, R., Felber, P., Pasin, M., Schiavoni, V., & Fetzer, C. (2018, July). SGX-aware container orchestration for heterogeneous clusters. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)* (pp. 730-741). IEEE. https://doi.org/10.48550/arXiv.1805.05847

[12] Vayghan, L. A., Saied, M. A., Toeroe, M., & Khendek, F. (2019). Kubernetes as an availability manager for microservice

applications. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1901.04946

[13] Wang, C., Kanso, A., Costache, S. V., Youssef, A. S., & Steinder, M. (2018, November 29). *US10915369B2 - Reward-based admission controller for resource requests in the cloud - Google Patents*. https://patents.google.com/patent/US10915369B2/en