WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Security and Privacy Challenges in Multi-Tenant Cloud Environments

Aksheev Tanwar[1]*, Varad Shete[2], Ritisha Kaushik[3], Om Deshpande[4], Reshma Sonar[5]

[1] Dept. of Computer Engineering and Technology Dr. Vishwanath Karad MIT World Peace University Pune, India
aksheev.tanwar@mitwpu.edu.in
[3] *Dept. of Computer Engineering and Technology Dr. Vishwanath Karad MIT World Peace University* Pune, India
ritisha.kaushik@mitwpu.edu.in
[5] *Dept. of Computer Engineering and Technology Dr. Vishwanath Karad MIT World Peace University* Pune, India
reshma.sonar@mitwpu.edu.in
[2] *Dept. of Computer Engineering and Technology Dr. Vishwanath Karad MIT World Peace University* Pune, India
varad.shete@mitwpu.edu.in
[4] *Dept. of Computer Engineering and Technology Dr. Vishwanath Karad MIT World Peace University* Pune, India
om.deshpande@mitwpu.edu.in

ABSTRACT—

Multi-tenant cloud environments enable cost- efficient resource sharing but introduce unique security and privacy challenges. When multiple customers share infrastruc- ture, vulnerabilities in one tenant or the cloud platform can compromise others. This paper analyzes such challenges through both theoretical and industry perspectives. We survey threat categories (isolation failures, side-channels, misconfigurations, identity risks) and examine real-world case studies on AWS, Azure, and GCP. For example, a critical Azure Cosmos DB bug (ChaosDB) exposed thousands of databases [2], and an Azure AD misconfiguration ("BingBang") allowed external attackers to manipulate Bing's CMS [3]. We discuss platform-specific practices (AWS Nitro isolation [7], GCP's Identity and Access management) and evaluate industry mitigations (encryption, strict tenant isolation). The paper concludes with recommended strategies (zero-trust identity, confidential computing) and future directions to secure multi-tenant clouds against emerging threats.

Introduction

Cloud computing has revolutionized IT by allowing mul- tiple organizations (tenants) to share computing resources on demand. In a *multi-tenant cloud*, virtual machines or containers from different users may co-reside on the same physical hardware [8]. This model offers economies of scale and scalability, but also raises security and privacy concerns. Shared resources mean that a vulnerability in the hypervisor, container engine, or management plane could affect many tenants simultaneously. As Shringarputale et al. note, "public clouds are inherently multi-tenant: applications deployed by different parties (including malicious ones) may reside on the same physical machines and share various hardware resources" [1].

This paper examines the new threat landscape created by multi-tenancy. We first define the problem and survey re- lated research and frameworks. Then, we categorize common threat vectors such as co-residency attacks, side-channels, and misconfiguration. Our analysis emphasizes real-world cases on major cloud platforms: Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP). For each, we present incidents or academic studies illustrating how multi-tenancy can be exploited. For instance, Azure's *ChaosDB* incident (a Cosmos DB vulnerability) gave attackers unrestricted access to thousands of databases [2]. On AWS, attacks on container and VM isolation have been demonstrated [5], [17]. On GCP, researchers found novel cross-tenant iden- tity and side-channel exfiltration paths [6]. We conclude by discussing mitigation strategies (strong isolation, encryption, continuous monitoring) and future directions (confidential computing, improved IAM practices) to strengthen security and privacy in cloud multi-tenancy.

Background and Related Work

Multi-tenancy is a foundational feature of cloud services. Early work by Varadarajan et al. demonstrated that cloud placement is often predictable, enabling an attacker to co- locate with a victim's VM on AWS, Azure, or Google Cloud [8]. Subsequent research confirmed that containerized workloads remain vulnerable: Shringarputale et al. showed that even modern container orchestration (Kubernetes on AWS/Azure) is prone to high-rate co-residency attacks, with over 90% detection success [1]. These studies underline that shared physical resources (CPU caches, memory, interconnects) can leak information across tenants.

Side-channel attacks in multi-tenant clouds have been ex- tensively analyzed. For example, Intel's speculative-execution bugs (Meltdown/Spectre, L1TF) allow a malicious VM to read data from another VM on the same CPU [9]. Google noted that L1 Terminal Fault (L1TF) means "private data fragments loaded in the L1 cache can potentially be read by a different process or VM that shares access to the cache" [10]. Recent work by Zhao et al. (ASPLOS 2024) demonstrates practical cross-tenant cache attacks on public clouds: they recovered 81% of secret ECDSA bits from a victim container

in GCP's Cloud Run and noted that AWS and Azure products are likely equally vulnerable [6]. Such side-channels require careful miti- gation by providers (cache partitioning, disabling simultaneous multithreading) or by tenants (constant-time algorithms).

Cloud security frameworks (e.g., CSA Guidance, NIST SP 800-144) highlight multi-tenancy risks but often focus on high-level controls. The literature emphasizes data isolation: any flaw in virtual machine monitors, container runtimes, or orchestration can lead to cross-tenant data leakage [11]. In practice, compliance requirements (GDPR, HIPAA) force CSPs and tenants to implement access controls and encryption, but the opacity of cloud infrastructure complicates verification. Recent case studies (discussed later) show that even small misconfigurations can undermine these controls.

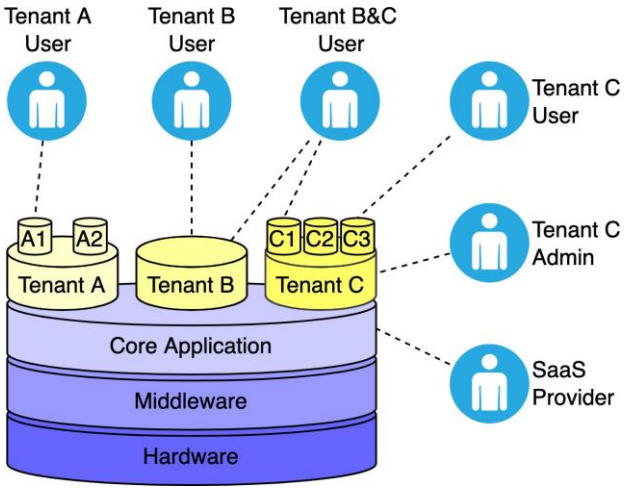


Fig. 1. General Multi-Tenant Cloud Architecture

Problem Definition

In a multi-tenant cloud, the fundamental security and pri- vacy problem is ensuring strict isolation between tenants who share underlying resources. We define the *multi-tenant security problem* as follows: attackers in one tenant or in the cloud infrastructure seek to gain unauthorized access to another tenant's resources (data, computation results) or to disrupt their availability. This encompasses:

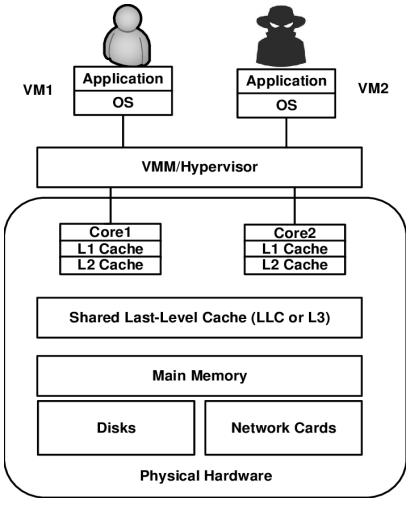
- **Isolation Failures**: Flaws in virtualization or container- ization that allow VM/container escapes, allowing one tenant to compromise the hypervisor or host OS. Example vectors include unpatched hypervisor bugs and container runtime vulnerabilities [5].
- **Side-Channel Leaks**: Covert channels via shared hard- ware (CPU cache, memory bus, GPU, FPGA, etc.) that let one tenant infer secrets from another [6].
- **Misconfiguration and Credential Risks**: Incorrect iden- tity or permission settings can inadvertently expose data across tenants. For instance, cloud identity misconfigura- tions may allow one tenant's account to access another's resources (see Section V).
- **Shared-Service Vulnerabilities**: Multi-tenant platforms (managed databases, Kubernetes control planes, etc.) may have bugs. A vulnerability in a shared service can impact all tenants using that service.
- **Regulatory and Privacy Concerns**: Tenants' data may have different compliance needs. Co-location can raise questions about data residency, key control, and auditabil- ity.

Analysis of Multi-Tenancy Threats

We analyze the above threat categories and discuss how they manifest in practice.

Co-residency and Side-Channel Attacks

Fig. 2. Side-Channel Attack Exploiting Shared Cache



Co-residency attacks aim to confirm that an attacker's VM/container is on the same host as a target's, enabling side channels. Modern hypervisors use random placement and live
migration to thwart co-location detection, but research shows attacks still succeed at scale. For example, Shringarputale et al. demonstrated attacker-controlled Docker containers on AWS/EKS could reliably detect and co-locate with victim con- tainers, achieving over 90% success [1]. Once co-located, the attacker can exploit microarchitectural side-channels: cache probing (Prime+Probe, Flush+Reload) or memory bus con- tention allow leakage of cryptographic keys and data from the victim.

The impact is broad: Zhao et al. implemented a cross- tenant cache attack on GCP Cloud Run, extracting ECDSA nonces from a victim service [6]. They observed that public clouds (AWS, Azure, GCP) use Intel CPUs with shared L3 caches, making them inherently susceptible. Indeed, multi-core side-channel vulnerabilities continue to be discovered (e.g., AMD SEV vulnerabilities), forcing cloud providers to deploy firmware and hypervisor patches. To mitigate these risks, AWS designed its Nitro hypervisor with strict isolation: Nitro mini- mizes shared hardware by offloading I/O and management to dedicated co-processors [7]. AWS documentation emphasizes that Nitro's goal is "strong isolation between security domains and limit the sharing of critical system resources across customers" [12]. GCP and Azure offer similar controls (e.g. GCP's Confidential VMs using AMD SEV, Azure's dedicated host offerings).

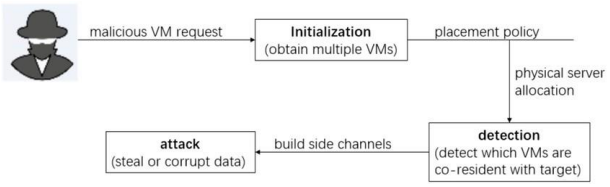


Fig. 3. Co-residency attack flow in a multi-tenant environment. The attacker achieves co-location, performs resource usage monitoring, and extracts data using side-channel techniques.

Virtualization and Container Escapes

Even with side-channel defenses, logical escapes from VMs or containers are critical threats. CVE-2019-5736 (runc container escape) exemplifies how a single flaw can impact multi-tenant container hosts. In this bug, a malicious container could overwrite the host's `/run/runc` binary, gaining root on the host and potentially on other containers [5]. AWS patched ECS to prevent this exploit, but such issues illustrate the danger of relying on up-to-date isolation mechanisms. Similarly, hypervisor vulnerabilities (e.g., past Xen or VMware bugs) can allow a tenant VM to compromise the host or other VMs. Providers conduct rigorous patching and sandboxing, but zero-day escapes remain a concern. For instance, research shows that even advanced container runtimes (Kata, gVisor) may not fully eliminate side-channels or escapes [13].

Identity and Access Misconfigurations

Multi-tenant clouds rely heavily on Identity and Access Management (IAM). Misconfigurations here can break tenant

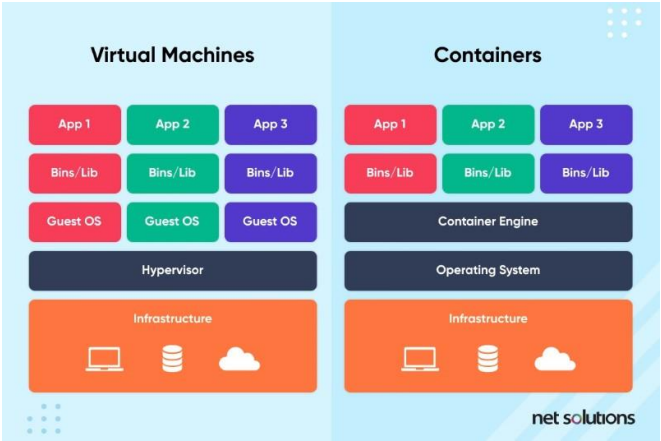


Fig. 4. Comparison of attack vectors in virtualization (VM) and container environments. Containers share the same OS kernel, increasing escape risks.

isolation. A notable case is the "BingBang" incident: Wiz researchers found that many Azure App Service applications were configured as multi-tenant without proper token vali- dation. As a result, one could obtain an OAuth token from any Azure tenant and access Microsoft's internal apps. In 25% of apps tested, attackers could use this flaw. Specifically, they manipulated Bing.com's content management system, altering search results and executing XSS on users [3]. This shows how an identity-layer oversight in a cloud platform can expose cross-tenant data. Azure and AWS provide features for restricting multi-tenant apps (e.g. tenant IDs in tokens), but developers often misinterpret shared responsibility, leading to risky exposures [14]. GCP's equivalent would be mis-set IAM bindings or enabled "allUsers" buckets that inadvertently share data between projects.

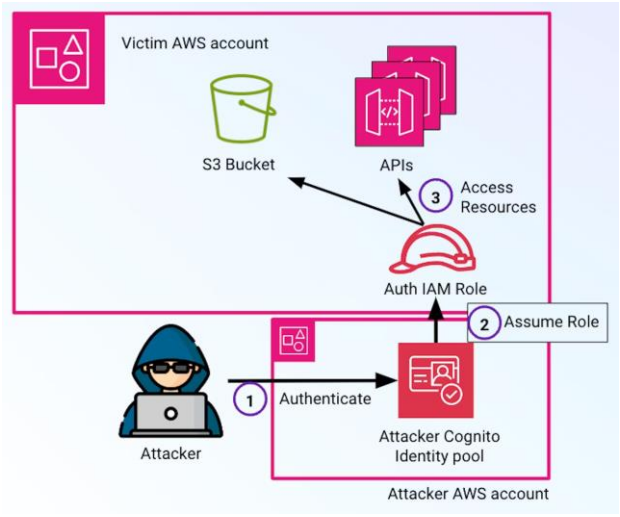


Fig. 5. IAM misconfiguration exploitation. Over-permissive roles or exposed OAuth tokens can lead to privilege escalation in cloud environments.

Shared-Service Vulnerabilities

Cloud providers host many managed services that are multi- tenant by nature (databases, function-as-a-service, etc.). A vulnerability in these services can be catastrophic. The 2021 ChaosDB event is a prime example [2]. An attacker chained flaws in Azure Cosmos DB's Notebook feature to steal or manipulate any customer's database keys and data. Hundreds of Azure accounts (including Fortune 500 companies) were affected before the issue was patched. Critically, tenants had no role in the configuration error – it was a provider service flaw. Similarly, AWS and GCP have had critical service vul- nerabilities (e.g., privilege escalations in management APIs) that could impact multiple tenants. These incidents highlight that multi-tenancy shifts some trust to the provider: even if a tenant is fully secure, a bug in a shared service can break isolation.

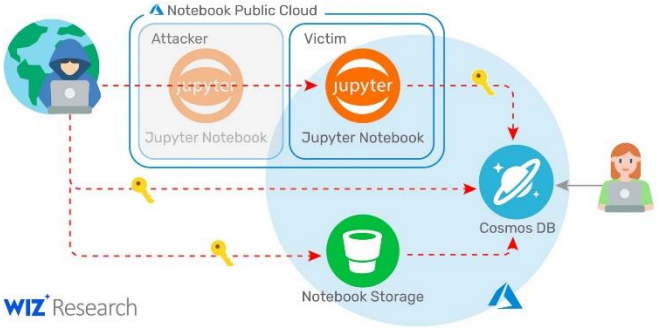


Fig. 6. Exploit chain of shared-service vulnerability (ChaosDB). Exploiting a notebook service exposed internal keys, enabling full control over the Cosmos DB instance.

Privacy and Compliance

Privacy in a multi-tenant cloud encompasses data protection and regulatory compliance. Unlike on-premises or single- tenant models, customers in a multi-tenant cloud often lack direct visibility into where their data resides or who might have access (even within the provider's organization). This can conflict with laws like GDPR or industry regulations (HIPAA, PCI-DSS). For instance, if two tenants are in different legal jurisdictions, data sovereignty issues arise. Encryption (at-rest and in-transit) is a standard mitigation, but keys are often managed by the provider unless the tenant uses client-side en- cryption. Emerging techniques like homomorphic encryption or secure enclaves (AWS Nitro Enclaves, Azure Confidential Computing) aim to allow computation on encrypted data, pre- serving privacy. However, they are not yet universally adopted. Additionally, monitoring and auditing in a multi-tenant cloud is challenging: tenants must rely on provider logs and third- party cloud security tools to demonstrate compliance. Overall, the privacy challenge is that cloud multi-tenancy requires both technological safeguards and strong policy controls to ensure tenant data is not inadvertently exposed [15].

Case Studies

We now examine concrete examples on major platforms to illustrate these challenges.

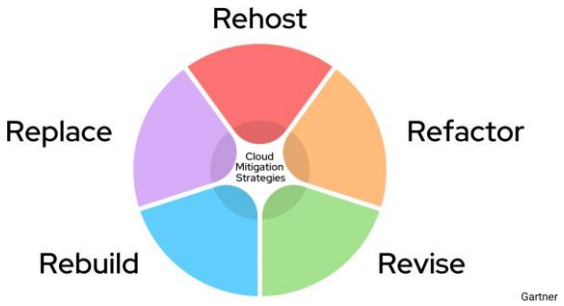


Fig. 7. Mitigation Strategies for Multi-Tenant Cloud Risks

AWS Case Studies

Amazon's AWS is the largest public cloud, and its scale invites scrutiny. AWS's Nitro system embodies the current best practice: a minimal hypervisor (really a microkernel) and dedicated hardware for I/O and management. AWS touts Ni- tro's isolation as a defense-in-depth measure [7]. Nonetheless, researchers show that side-channels remain feasible on AWS. Shringarputale et al. validated cache attacks on Amazon EC2 and EKS containers with high success [1]. They observed that noise (from other tenants) does not fully prevent co- residency attacks. The aforementioned runc container escape (CVE-2019-5736) affected AWS container services [5]. Al- though AWS patched it rapidly, it underscores that multi-tenant container hosts must be promptly updated.

On the misconfiguration front, many data breaches have involved AWS, often not due to multi-tenancy but due to user error (public S3 buckets, weak IAM policies). While these are not novel multi-tenant threats, they reflect the risk of broad access in a shared environment. AWS offers tools like IAM Access Analyzer and AWS Config to help tenants audit their multi-account setups and avoid accidental cross-account access.

Azure Case Studies

Microsoft Azure has had notable multi-tenancy incidents. The **ChaosDB** vulnerability (August 2021) is one of the most severe: a flaw in Cosmos DB allowed attackers to list and modify the primary keys of any Cosmos database, affecting thousands of customers [2]. The attackers achieved full data access and the ability to manipulate services. Microsoft labeled it "the worst" Azure breach seen. This incident stemmed from a combination of over-permissive Jupyter Notebook integra- tion and a missing token validation step in the Cosmos DB management plane. It demonstrates how a platform service's design error can break multi-tenant security.

Another Azure-specific case is **BingBang** (March 2023) [3]. Wiz researchers discovered that Microsoft's own multi-tenant Azure AD applications were misconfigured. Specifically, an

internal Bing CMS allowed any Azure AD user (from any ten- ant) to bypass authentication. Exploiting this, the researchers could change Bing's search results and inject scripts into users' browsers. The root cause was confusion over Azure AD's "multi-tenant app" setting: Azure did not restrict app users to a specific tenant, and the CMS did not validate token issuer. This attack path was purely identity-based – the code was multi- tenant by choice – and affected user privacy across tenants. Microsoft issued patches after being notified.

Google Cloud Case Studies

GCP has fewer publicized breaches but presents its own multi-tenant challenges. In 2024, Data Center Knowledge highlighted *cross-tenant and multi-tenant threats* in Google Cloud [4]. They note that typical IAM configurations assume a single-organization boundary; however, when cloud identities span personal and corporate accounts, data can leak across tenants (e.g., an engineer copying corporate data to a personal GCP project). To address this, Google introduced "Princi- pal Access Boundary" (PAB) to restrict service accounts to certain projects, limiting cross-tenant moves [16]. This is a direct response to multi-tenant risk scenarios where an insider or compromised account could exfiltrate data across tenant boundaries.

On the research side, Zhao et al. (2024) demonstrated a potent attack in GCP: by carefully co-locating and performing a cache attack in Cloud Run, they recovered an applica- tion's cryptographic key bits [6]. The attack chained container scheduling and cache timing, showing that "multi-tenant cloud products from other vendors, such as AWS and Azure, may also be susceptible" [6]. This work underscores that even GCP's serverless offerings (Cloud Run, App Engine) can be vulnerable to multi-tenant side channels. Google's mitigations include disabling SMT (hyper-threading) on some instances and offering confidential VMs, but fully preventing such leaks remains an active area.

TABLE I

COMPARISON OF CLOUD PROVIDERS ON MULTI-TENANCY SECURITY

Provider	Isolation Method	Side-Channel Protection	Data Encryption
AWS	Nitro	Yes	Full
Azure	Hyper-V	Partial	At-rest
GCP	gVisor	Yes	Full

TABLE II

COMPARISON OF MULTI-TENANT THREATS ACROSS CLOUD PROVIDERS

Threat Type	AWS	Azure	GCP
Side-channels	Present (Nitro mitigates)	Present	Present (Cloud Run case)
Identity risks	IAM misconfigs	BingBang	Cross-project leaks
Shared-service flaws	Less reported	ChaosDB	Minimal but possible

Discussion

The case studies and analysis above reveal patterns and open questions. First, no platform is immune: AWS, Azure, and GCP each face multi-tenancy issues of their own design. Providers invest heavily in isolation (e.g., AWS Nitro, Azure Hyper-V, GCP's Capsule VM micro-VMs) but researchers continue to find innovative exploits. This cat-and-mouse dy- namic suggests the importance of layers of defense.

Key lessons include:

- **Least Privilege and Zero Trust**: Tenants should assume other tenants may be compromised. Implement least- privilege IAM, network segmentation (VPCs, private end- points), and continuous monitoring (AWS GuardDuty, Azure Security Center, GCP Security Command Center). Zero-trust network models, where workloads authenticate to each other, mitigate lateral movement in a breached host.
- **Confidential Computing**: Utilizing hardware enclaves (Intel SGX, AMD SEV, AWS Nitro Enclaves, GCP Confidential VMs) can protect data-in-use even if the host is compromised. If sensitive workloads run inside an enclave, even a co-resident attacker gains little. Azure and AWS offer such features, though they limit compute types and performance.
- **Formal Assurance of Isolation**: Providers should con- tinue to harden hypervisors and formally verify criti- cal components. Nvidia's forthcoming confidential GPU computing and forthcoming RISC-V secure processors may improve trust. Meanwhile, tenants should follow provider advisories (e.g., Intel microcode patches) and avoid collocating highly sensitive workloads with un- known tenants.
- **Regulatory Controls**: Organizations must negotiate SLAs and compliance attestation regarding multi-tenancy. If standards demand physical isolation (e.g. for certain healthcare or government data), using dedicated hosts or region filtering is necessary.
- **Awareness and Training**: Many multi-tenant risks come from human error (misconfigurations). The BingBang case shows even Microsoft's engineers fell for a multi- tenant trap [3]. Cloud teams must be educated about shared-responsibility: for multi-tenant apps, they must enforce their own tenant restrictions.

While industry tools (CSP-specific controls, CSPM, SIEM) help detect and prevent misconfigurations, research challenges remain. Future work may explore secure multi-party computa- tion for joint workloads, improved detection of co-residence, or AI-driven intrusion detection specialized for multi-tenant telemetry.

Conclusion

Multi-tenant cloud environments offer great benefits but come with complex security and privacy challenges. This paper has identified major threat categories: cross-tenant side- channels, VM/container escapes, identity misconfigurations, and shared-service vulnerabilities. Through analysis and case studies on AWS, Azure, and GCP, we saw that even lead- ing cloud providers can be vulnerable. Real incidents like ChaosDB [2] and BingBang [3] highlight the impact of such flaws.

To protect multi-tenant clouds, both providers and tenants must adopt stringent isolation and monitoring. Providers con- tinue to evolve their architectures (e.g., Nitro, confidential computing, Principal Access Boundaries) [7], [16]. Tenants must encrypt sensitive data, apply least privilege IAM, and stay informed of cloud-security best practices. Finally, collab- oration between research and industry is key: sharing lessons from each breach or vulnerability will improve defenses across all platforms. As cloud adoption grows, ensuring privacy and security in multi-tenancy will remain an urgent, evolving challenge.

Acknowledgments

The authors thank the researchers and practitioners whose work and disclosed incidents have informed this survey.

REFERENCES

[1] S. Shringarputale, P. McDaniel, K. Butler, and T. La Porta, "Co- residency Attacks on Containers are Real," in *Proc. ACM CCSW*, 2020, pp. 8:1–8:10.

[2] N. Ohfeld and S. Tzadik, "ChaosDB: How we hacked thousands of Azure customers' databases," Wiz Blog, Aug. 26, 2021. [Online]. Available: https://www.wiz.io/blog/ chaosdb-how-we-hacked-thousands-of-azure-customers-databases

[3] H. Ben-Sasson, "BingBang: AAD misconfiguration led to Bing.com results manipulation and account takeover," Wiz Blog, Mar. 29, 2023. [Online]. Available: https://www.wiz.io/blog/ azure-active-directory-bing-misconfiguration

[4] K. Haller, "Multi-Tenant and Cross-Tenant Threats in Google Cloud and Beyond," *Data Center Knowledge*, Sept. 25, 2024. [Online]. Available: https://www.datacenterknowledge.com/cybersecurity/ multi-tenant-and-cross-tenant-threats-in-google-cloud-and-beyond

[5] A. Kumar, "Anatomy of CVE-2019-5736: A runc container escape!," *AWS Compute Blog*, Apr. 4, 2019. [Online]. Available: https://aws.amazon.com/blogs/compute/ anatomy-of-cve-2019-5736-a-runc-container-escape/

[6] Z. Zhao et al., "Last-Level Cache Side-Channel Attacks Are Feasible in the Modern Public Cloud," in *Proc. ACM ASPLOS*, 2024, pp. 1–15.

[7] Amazon Web Services, "The Security Design of the AWS Nitro System," AWS Whitepaper, 2018. [Online]. Available: https://docs.aws.amazon. com/whitepapers/latest/security-design-of-aws-nitro-system/

[8] V. Varadarajan et al., "Resource-Freeing Attacks: Improve Your Cloud Performance (at Your Neighbor's Expense)," in *Proc. IEEE S&P*, 2015, pp. 423–438.

[9] Intel Security, "Analyzing Potential Bounds Check Bypass Vulnerabili- ties," Intel White Paper, 2018.

[10] Google Cloud Security Team, "Mitigating CPU Vulnerabilities in Google Cloud," Technical Report, 2018.

[11] NIST, "Guidelines on Security and Privacy in Public Cloud Computing," SP 800-144, 2011.

[12] AWS, "AWS Nitro System Documentation," 2023. [Online]. Available: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/nitro-system. html

[13] M. Research, "Container Runtime Security: Challenges and Solutions," in *USENIX Security*, 2023.

[14] Microsoft, "Azure AD Multi-Tenant Application Best Practices," 2023. [Online]. Available: https://learn.microsoft.com/en-us/azure/active-directory/develop/howto-convert-app-to-be-multi-tenant

[15] EU GDPR, "Cloud Computing Guidelines," 2023. [Online]. Available: https://gdpr.eu/cloud-computing/

[16] Google Cloud, "Principal Access Boundaries Documentation," 2024. [Online]. Available: https://cloud.google.com/iam/docs/ principal-access-boundaries

[17] AWS Security Team, "Container Isolation in AWS ECS and EKS", AWS Security Blog, 2022. [Online]. Available: https://aws.amazon.com/blogs/ security/container-isolation-best-practices/