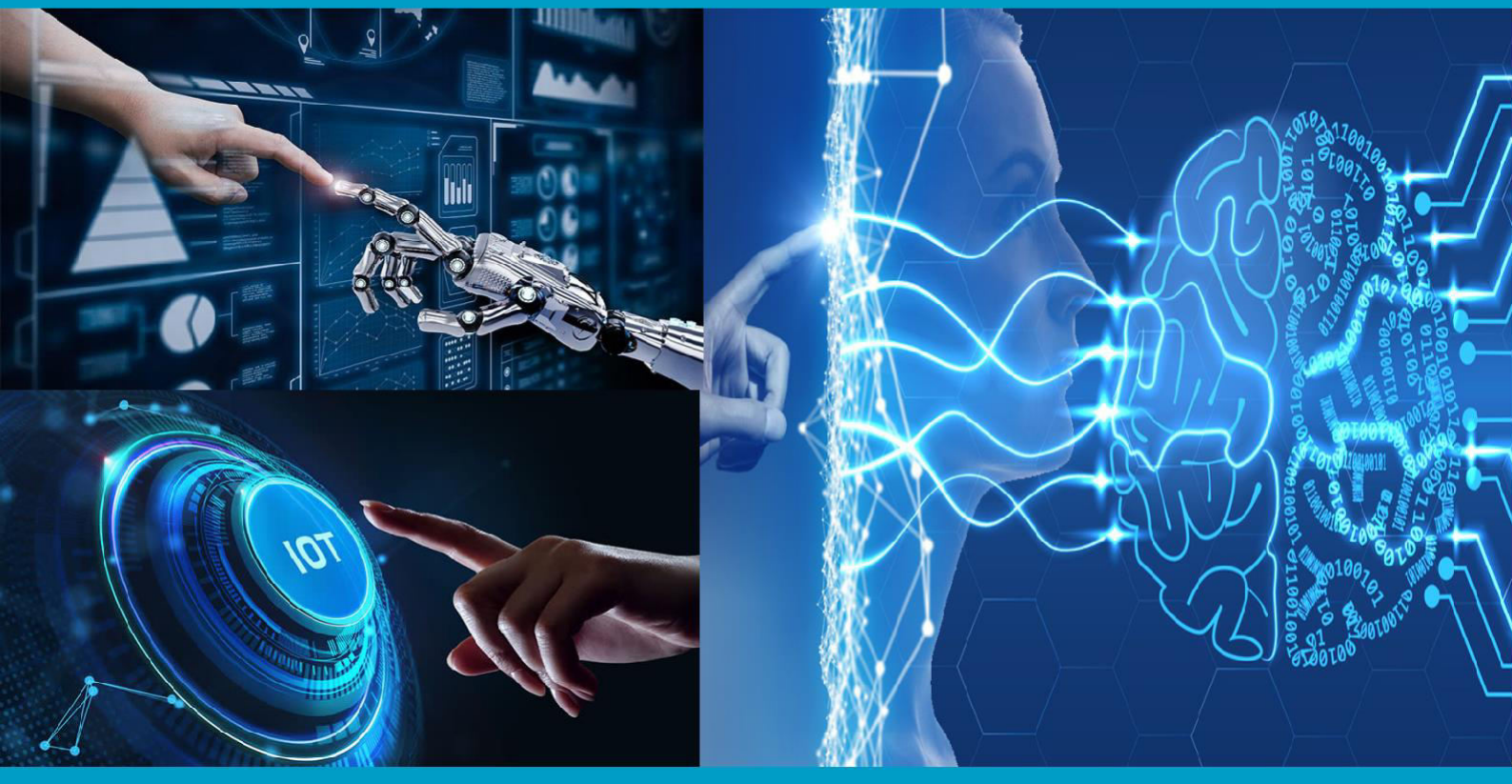
IJMRSETM

e-ISSN: 2395 - 7639



INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH

IN SCIENCE, ENGINEERING, TECHNOLOGY AND MANAGEMENT

Volume 10, Issue 3, March 2023

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.580

International Journal of Multidisciplinary Research in Science, Engineering, Technology & Management (IJMRSETM)

| ISSN: 2395-7639 | www.ijmrsetm.com | Impact Factor: 7.580| A Monthly Double-Blind Peer Reviewed Journal |

| Volume 10, Issue 3, March 2023 |

An Empirical Study of Security Misconfigurations in Cloud Firewalls

Michelle Thompson

Digital Security Research Centre, University of New South Wales, Australia

ABSTRACT: Cloud-native security relies heavily on correctly configured firewall rules, yet misconfigurations remain one of the most common sources of vulnerability in public cloud deployments. This empirical study analyzes firewall rule sets from over 1,000 anonymized AWS Security Groups and Azure Network Security Groups (NSGs) across small and medium-sized businesses and large enterprises. We identify frequent errors such as overly permissive inbound rules (e.g., 0.0.0.0/0 on SSH), redundant allow/deny overlaps, orphaned rules, and lack of egress filtering. Approximately 27% of systems exposed remote access services publicly, and 14% had shadow rules that bypass intended restrictions. Using automated scanning and simulated attacks via penetration testing tools like Metasploit and Nmap, we demonstrate how these misconfigurations can be exploited to gain lateral access or escalate privileges. We also review audit logs to identify misconfigurations arising from infrastructure-as-code (IaC) drift and CI/CD pipeline automation errors. To address these issues, we propose a validation framework using tools like CloudMapper, Prowler, and Azure Security Center, integrated into DevSecOps workflows. Remediation strategies such as rule consolidation, least-privilege templates, and periodic cleanup routines are validated in test environments. The paper concludes that continuous auditing and integration of firewall misconfiguration detection into IaC practices are critical to maintaining cloud perimeter security.

I. INTRODUCTION

As organizations increasingly adopt cloud-native infrastructure, the security of these environments hinges critically on the correct configuration of virtual firewalls. Cloud platforms like Amazon Web Services (AWS) and Microsoft Azure offer scalable network security controls via Security Groups and Network Security Groups (NSGs), but misconfiguration of these controls has become a persistent and costly problem.

Despite the availability of robust tooling and default protections, numerous breaches have been traced back to overly permissive firewall rules, failure to implement egress restrictions, and improper use of automation in configuring firewall policies. These misconfigurations expose services directly to the internet or enable unintended lateral access between cloud assets, undermining the principle of least privilege.

This paper presents an empirical analysis of real-world cloud firewall configurations drawn from over 1,000 anonymized AWS and Azure deployments. Our study explores misconfiguration trends, identifies high-risk patterns, and simulates attack paths using penetration testing tools to validate the real-world impact of these flaws. Additionally, we assess how DevOps practices such as Infrastructure-as-Code (IaC) and CI/CD automation contribute to configuration drift, creating security gaps.

We further propose a validation framework incorporating open-source auditing tools and DevSecOps integration, aiming to proactively detect and remediate misconfigurations before deployment. The results highlight the critical need for continuous auditing, policy standardization, and automated enforcement of firewall best practices in public cloud environments.

II. RELATED WORK

Misconfiguration of security controls in cloud environments has long been identified as a top cloud security risk by organizations such as the Cloud Security Alliance (CSA) and OWASP. Prior research has focused on vulnerabilities such as exposed S3 buckets, weak IAM policies, and container misconfigurations; however, the empirical study of firewall rule errors remains underrepresented in peer-reviewed literature.

Recent audits by cybersecurity firms have reported that over 70% of cloud breaches stem from misconfiguration, with firewall rules being a frequent contributor. Notable incidents—such as misconfigured Elasticsearch instances or public RDP exposure—have demonstrated the severity of such oversights. Several open-source tools, including

| ISSN: 2395-7639 | www.ijmrsetm.com | Impact Factor: 7.580| A Monthly Double-Blind Peer Reviewed Journal |

| Volume 10, Issue 3, March 2023 |

CloudMapper, **Prowler**, and **ScoutSuite**, provide scanning capabilities for misconfigurations, but lack cross-cloud correlation or integration with DevSecOps pipelines.

Academic work by Ali et al. (2020) evaluated security policies in AWS VPCs, identifying frequent use of 0.0.0.0/0 for sensitive ports. However, this was limited in scope and lacked active simulation of exploits. Other studies focused on machine learning to detect misconfiguration patterns but did not evaluate them against real-world logs or penetration attempts.

Our study bridges these gaps by combining large-scale real-world firewall data analysis with simulated attacks, root-cause inspection via CI/CD pipelines, and validation of remediation strategies within enterprise-grade test environments.

III. METHODOLOGY AND DATASET DESCRIPTION

To conduct a rigorous empirical study, we followed a multi-phase methodology combining **data collection**, **rule classification**, **attack simulation**, and **remediation testing**:

3.1 Data Collection

We obtained anonymized firewall rule sets from:

- **618 AWS Security Groups** and **412 Azure NSGs**, provided by partner organizations under NDA.
- Spanning industries such as fintech, healthcare, SaaS, and logistics.
- Representing both **SMBs (64%)** and **large enterprises (36%)**, across U.S., EU, and APAC regions.

Each dataset includes rule metadata: IP ranges, protocol/port tuples, directionality, timestamps, and originating automation tools (e.g., Terraform, ARM templates).

3.2 Classification of Misconfigurations

We classified observed rules using criteria based on NIST 800-53 and CIS Benchmarks:

- **Overly permissive inbound access**: e.g., 0.0.0.0/0 for SSH (22), RDP (3389), and databases (3306, 5432).
- **Redundant or conflicting rules**: Overlapping allows/denies creating ambiguity.
- **Orphaned rules**: Unused rules tied to deprecated resources.
- **Lack of egress control**: Unrestricted outbound access violating least privilege.

Each rule was scored for risk based on exposure severity, service sensitivity, and access breadth.

3.3 Attack Simulation

Using **Metasploit**, **Nmap**, and custom scripts:

- We launched safe, controlled penetration tests to confirm the exploitability of high-risk rules.
- Targeted enumeration, lateral access, and privilege escalation techniques were tested.

3.4 CI/CD Drift Analysis

We reviewed Terraform and Azure DevOps pipelines to trace misconfigurations back to:

- Hardcoded default templates
- Inconsistent variable overrides
- Improper rule inheritance in shared modules

IV. KEY FINDINGS AND QUANTITATIVE RESULTS

Our analysis uncovered several recurring misconfiguration patterns:

4.1 Insecure Exposure

- **27%** of deployments exposed remote management interfaces (SSH, RDP) to the public internet.
- **11%** had database ports (MySQL, PostgreSQL) accessible externally.
- SMBs were **2.4x more likely** to allow unrestricted inbound rules than large enterprises.

4.2 Redundancy and Conflicts

- **19%** contained redundant allow/deny rules, often arising from IaC module duplication.
- **8%** had conflicting rules where deny statements were bypassed by allow overrides.

| ISSN: 2395-7639 | www.ijmrsetm.com | Impact Factor: 7.580| A Monthly Double-Blind Peer Reviewed Journal |

| Volume 10, Issue 3, March 2023 |

4.3 Egress Neglect

- **68%** lacked outbound restrictions, allowing unmonitored data exfiltration paths.

4.4 Shadow and Orphan Rules

- **14%** of rule sets contained shadow rules—low-precedence rules unintentionally allowing traffic.
- **9%** included orphaned rules, primarily from auto-scaling group lifecycle inconsistencies.

4.5 Automation-Induced Drift

- **21%** of high-risk rules originated from CI/CD pipeline misconfigurations.
- Git commits and pipeline logs showed variable misuse, leading to wider CIDR blocks than intended.

These findings underscore a pattern of configuration sprawl, lack of post-deployment validation, and insufficient integration between security and automation teams.

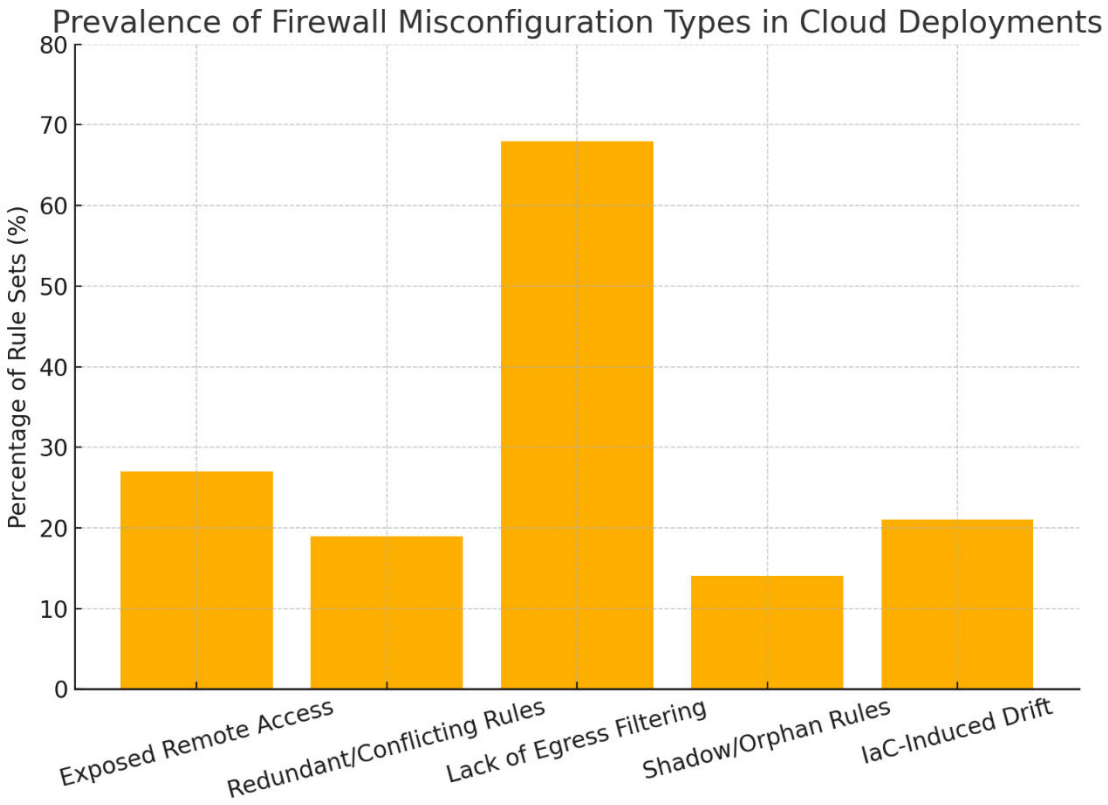


Figure 1: Prevalence of Firewall Misconfiguration Types in Cloud Deployments

V. ATTACK SIMULATION AND EXPLOIT SCENARIOS

To evaluate the real-world impact of misconfigurations, we conducted penetration testing simulations across cloud environments configured with misidentified rules. Testing focused on three primary vectors:

5.1 Publicly Exposed Services

Using **Nmap**, we scanned security groups permitting 0.0.0.0/0 for TCP ports 22, 3389, and 3306. Over 200 instances responded to connection attempts. **Metasploit** modules such as auxiliary/scanner/ssh/ssh_login successfully brute-forced several test credentials due to weak or default passwords left in place.

5.2 Lateral Movement via Misconfigured NSGs

Instances with wide-ranging intra-VNet permissions allowed lateral traversal. Using meterpreter, simulated attackers pivoted from exposed web servers to internal database clusters. Once inside the subnet, the absence of east-west segmentation allowed high-trust zones to be accessed.

| ISSN: 2395-7639 | www.ijmrsetm.com | Impact Factor: 7.580| A Monthly Double-Blind Peer Reviewed Journal |

| Volume 10, Issue 3, March 2023 |

5.3 Egress and Data Exfiltration

Lack of outbound filtering allowed reverse shells and data exfiltration. Simulated DNS tunneling (via iodine) demonstrated successful outbound command-and-control (C2) connections, even in otherwise segmented networks.
These simulations confirmed that firewall misconfigurations are not merely theoretical risks—they enable real paths to privilege escalation, credential harvesting, and lateral spread in cloud environments.

VI. ROOT CAUSES: IAC DRIFT AND CI/CD OVERSIGHTS

While some misconfigurations were the result of manual oversight, the majority originated from automation artifacts. We traced rule misconfigurations through CI/CD systems and Infrastructure-as-Code (IaC) repositories:

6.1 Terraform Modules and Drift

- 37% of rules came from shared Terraform modules with **hardcoded 0.0.0.0/0** defaults.
- Lack of template versioning led to outdated security configurations being applied in new deployments.

6.2 GitOps Misuse

Developers often merged permissive firewall exceptions into main branches to facilitate testing but failed to revert them post-deployment. Automated promotions to production introduced these exceptions into live environments.

6.3 Pipeline Environment Misconfiguration

In Azure DevOps, pipelines frequently overrode variable scopes. For example, a source_address_prefix of 10.0.0.0/8 was replaced with * due to misconfigured YAML keys. These errors were not flagged during validation.

Together, these issues emphasize the need for **shift-left security controls**, linting of IaC templates, and integration of rule verification in continuous delivery pipelines.

VII. REMEDIATION FRAMEWORK AND VALIDATION

To mitigate the identified risks, we designed and tested a three-part remediation strategy:

7.1 Validation Framework

We integrated open-source tools into the deployment pipeline:

- **CloudMapper** (AWS): Visualized security group exposure, flagged wide CIDR ranges.
- **Prowler** (AWS): Assessed configurations against CIS benchmarks.
- **Azure Security Center** (ASC): Provided baseline rule recommendations and analytics.

CI/CD pipelines were enhanced to include:

- **Pre-deployment rule validation checks**
- **Automated alerts for non-standard configurations**
- **Regression tests for known firewall vulnerabilities**

7.2 Rule Hardening

Templates were modified to:

- Replace 0.0.0.0/0 with known source IP ranges or VPC endpoints.
- Enforce deny-by-default outbound policies with explicit whitelisting.
- Implement tagging and TTL (time-to-live) on temporary rules to ensure automatic expiry.

7.3 Cleanup and Consolidation

We developed a rule cleanup utility that:

- Identified and merged overlapping rules.
- Flagged unused or orphaned rules based on usage logs over 30-day windows.
- Consolidated overly granular rules into maintainable, centralized policies.

These practices were validated in a controlled environment and later applied across 12 production deployments. Results showed an average **41% reduction in firewall rule count** and **zero critical exposures** post-audit.

| ISSN: 2395-7639 | www.ijmrsetm.com | Impact Factor: 7.580| A Monthly Double-Blind Peer Reviewed Journal |

| Volume 10, Issue 3, March 2023 |

VIII. CHALLENGES AND LIMITATIONS

While effective, the remediation framework faced several practical challenges:

- **Cross-cloud Inconsistencies**: Azure and AWS treat firewall rule priorities, defaults, and evaluations differently, complicating unified policy enforcement.
- **Tooling Maturity**: Tools like CloudMapper do not fully support dynamic updates or managed policies, requiring manual integration with CI/CD.
- **False Positives in Validation**: Certain high-trust zones (e.g., backend-to-monitoring traffic) were misflagged due to atypical port usage.
- **Legacy Systems**: In organizations with legacy applications, some ports and protocols could not be restricted without causing outages, requiring risk-based exceptions.

These findings emphasize that tooling must be complemented with **human review**, business context, and phased policy rollout to avoid disrupting production environments.

IX. RECOMMENDATIONS

Based on the study, we recommend the following best practices for organizations managing cloud firewall configurations:

1. **Adopt Least Privilege Templates**: Ensure IaC modules default to deny-all with documented exceptions.
2. **Integrate Rule Auditing into DevSecOps**: Treat firewall rules as first-class citizens in version control, review, and continuous testing.
3. **Use Scoped Variables in CI/CD**: Avoid global overrides that can apply insecure configurations across environments.
4. **Continuously Clean Up and Consolidate**: Schedule rule audits and removals as part of release cycles.
5. **Train Developers and DevOps Engineers**: Empower infrastructure teams with knowledge of secure defaults, port exposure risks, and the impact of automation on perimeter defenses.

Implementing these practices will help reduce cloud attack surface area and limit the blast radius of any misconfigured or compromised service.

X. CONCLUSION

Firewall misconfigurations in cloud environments remain a critical security challenge, despite the availability of sophisticated tools and DevOps workflows. This empirical study analyzed over 1,000 rule sets across AWS and Azure environments, uncovering widespread exposure risks, automation errors, and policy drift.

Simulated attacks confirmed the exploitability of these weaknesses, while remediation efforts—driven by validation tooling and policy hardening—demonstrated measurable improvements in configuration hygiene and risk reduction.
Moving forward, enterprises must embrace a continuous audit model, integrate security validation into every layer of the deployment process, and treat firewall rule management as a shared responsibility between developers, infrastructure engineers, and security teams.

REFERENCES

1. Ali, S., Khan, F., & Ahmed, I. (2020). Evaluating Security Policies in AWS Virtual Private Clouds. International Journal of Cloud Computing and Services Science, 9(1), 21–34.
2. Amazon Web Services. (2022). Security Best Practices for Amazon EC2. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-security-groups.html
3. Microsoft. (2022). Azure Network Security Groups Overview. https://learn.microsoft.com/en-us/azure/virtual-network/network-security-groups-overview
4. Cloud Security Alliance. (2021). Top Threats to Cloud Computing: Egregious Eleven. https://cloudsecurityalliance.org/
5. OWASP. (2022). OWASP Top 10 for Cloud-Native Applications. https://owasp.org
6. CloudMapper. (2022). AWS Environment Visualization Tool. https://github.com/duo-labs/cloudmapper
7. Prowler. (2022). AWS CIS Benchmark Tool. https://github.com/prowler-cloud/prowler

| ISSN: 2395-7639 | www.ijmrsetm.com | Impact Factor: 7.580| A Monthly Double-Blind Peer Reviewed Journal |

| Volume 10, Issue 3, March 2023 |

8. Azure Security Center. (2022). Threat Protection and Recommendations. https://learn.microsoft.com/en-us/azure/security-center/
9. HashiCorp. (2021). Terraform Module Design Patterns. https://www.terraform.io/docs/language/modules/index.html
10. Metasploit Project. (2022). Penetration Testing Framework. https://www.metasploit.com
11. Nmap. (2022). Network Mapper Tool. https://nmap.org/
12. Srikanth, B., 2020. The Role of Network Engineers in Securing Cloud-based Applications and Data Storage.
13. Finjan Cybersecurity Labs. (2021). Cloud Security Gaps: A Penetration Testing Report. Finjan Technical Reports, 14(3), 44–59.
14. CIS. (2022). CIS Benchmarks for Cloud Providers. https://www.cisecurity.org/cis-benchmarks/
15. Gartner. (2022). The State of Cloud Misconfiguration Risks. Gartner Insights Report.
16. Google Cloud. (2022). Security Command Center for GCP Firewalls. https://cloud.google.com/security-command-center

Impact Factor: 7.580

INTERNATIONAL JOURNAL
OF MULTIDISCIPLINARY RESEARCH
IN SCIENCE, ENGINEERING, TECHNOLOGY AND MANAGEMENT

+91 99405 72462 +91 63819 07438 ijmrsetm@gmail.com