Evaluating the Effectiveness of Azure Network Security Groups in Contained Lateral Threat Movement

Alphonse Joseph
Master in Cybersecurity / University West
alphonse.joseph@student.hv.se

***Abstract*—Cloud virtual machines are often deployed inside shared internal networks. While this simplifies management, it also creates opportunities for internal threats to spread once a single system is compromised. This project investigates whether Azure Network Security Groups can be used to limit such internal spread by blocking lateral communication between virtual machines. A small Azure environment was created using two Ubuntu Server virtual machines placed in the same subnet. Internal connectivity was tested before and after applying a restrictive subnet-level Network Security Group rule. The results show that unrestricted lateral communication exists by default, but is effectively blocked once the rule is applied. Network flow logs were used to confirm that traffic was denied at the network layer. The findings demonstrate that simple subnet-level segmentation provides a practical and effective method for reducing internal risk in cloud environments.**

***Index Terms*—Cloud security, lateral movement, Network Security Groups, Microsegmentation, Microsoft Azure**

I. Introduction

Cloud platforms make it easy to deploy and scale virtual machines, often placing multiple systems inside the same internal network. While this design improves efficiency, it also introduces security concerns. If one system is compromised, attackers can often communicate freely with other systems inside the same subnet. This internal spread, commonly referred to as lateral movement [1], is frequently used by automated malware and post-compromise attackers .

Preventing lateral movement is therefore an important goal in cloud security. Network segmentation and microsegmentation are widely recommended techniques for reducing internal attack surfaces and limiting such movement [2] . Azure provides Network Security Groups as a built-in mechanism for controlling network traffic between resources. This project examines how effective these security groups are when used at the subnet level to block internal communication. Rather than deploying real malware, the project uses safe connectivity tests to simulate worm-like behavior and evaluate containment

II. Experimental Setup

The experiment was carried out in Microsoft Azure using a single resource group. A virtual network was created with one subnet to represent a flat internal cloud environment. Two Ubuntu Server virtual machines were deployed within this subnet. One machine referred to as the JumpBox was assigned a public IP address and used to initiate internal connectivity tests. The second machine, referred as the target do not have a public IP address and was reachable only from within the virtual network. A Telnet service was enabled on the target machine to represent a weak internal service. A separate

TABLE I
Experimental Environment Configuration

Component	Description
Cloud Platform	Microsoft Azure
Virtual Network	Single VNet
Subnet	One subnet (flat internal network)
JumpBox VM	Ubuntu Server, Public IP enabled
Target VM	Ubuntu Server, No public IP
Internal Service	Telnet (Port 23)
Security Control	Subnet-level Network Security Group

Network Security Group was created and later attached to the subnet.Azure Network Security Groups provide rule-based filtering of inbound and outbound traffic at both subnet and network interface levels [3]. This security group was used to control all traffic between the virtual machines.

III. Methodology

An experimental approach was used. The independent variable in the experiment was the Network Security Group configuration, while the dependent variables were the results of internal connectivity tests.

The experiment followed two main phases. In the first phase, baseline testing was performed before any restrictive rules were applied. From the JumpBox, the target machine was tested using ping, Telnet, and safe Metasploit auxiliary scanning modules [4].These tests confirmed whether internal communication was allowed by default.

TABLE II
Tools Used in the Experiment

Tool	Purpose
Ping	Basic connectivity testing
Telnet	Internal service reachability test
Metasploit	Safe service scanning (auxiliary modules)
Azure NSG	Traffic filtering and segmentation
Network Watcher	Flow log monitoring and verification

Table II summarizes the tools used and their roles in the experimental workflow.

In the second phase, a subnet-level Network Security Group rule was applied to deny all traffic originating from and destined to the virtual network. The same tests were then repeated under identical conditions. Azure Network Watcher virtual network flow logs were enabled to provide independent confirmation of traffic blocking [3].

TABLE III
TEST SCENARIOS

Scenario	NSG Configuration	Expected Behavior
Baseline	No restrictive rules	Internal traffic allowed
Contained	Subnet deny rule applied	Internal traffic blocked

Automation was used only to ensure repeatability of tests, not to increase realism or impact

IV. RESULTS

Before the Network Security Group rule was applied, all internal connectivity tests succeeded. The JumpBox was able to reach the target machine using ping and Telnet and Metasploit scans detected open services on the target system. These results confirmed that lateral movement was possible within the subnet.

TABLE IV
INTERNAL CONNECTIVITY TEST RESULTS

Test	Before NSG	After NSG
Ping (ICMP)	Success	Blocked
Telnet (Port 23)	Success	Blocked
Metasploit Scan	Services detected	No response

After the subnet level deny rule was applied, all internal communication attempts failed. Ping requests timed out.Telnet connections could not be established, and Metasploit scans returned no results. This demonstrated that the Network Security Group effectively blocked internal traffic between the two machines.

Virtual network flow logs supported these observations. The logs showed denied traffic entries associated with the applied rule, confirming that traffic was blocked at the network layer rather than failing due to configuration errors.

TABLE V
FLOW LOG OBSERVATIONS AFTER NSG ENFORCEMENT

Observation	Result
Inbound internal traffic	Denied
Outbound internal traffic	Denied
Packets delivered to target	None
NSG rule matched	Subnet deny rule

V. DISCUSSION

The results clearly show that flat cloud networks allow unrestricted internal communication by default. Even simple services become reachable from any machine within the same subnet. This creates favorable conditions for lateral movement once a system is compromised.

Applying a subnet-level Network Security Group rule proved to be an effective containment measure. Similar approaches based on microsegmentation and zero trust principles have been widely discussed in recent research [5]. The solution required no changes to the operating systems and relied entirely on Azure's native network controls, highlighting the value of network-level segmentation as a basic but powerful security practice.

This study focuses on network-level containment and does not address identity-based or management-plane attacks. However, limiting east–west traffic still reduces the potential impact of many internal threats and helps contain compromise scope.

A. Threat Model Considerations

The threat model considered in this project assumes that an attacker has already gained initial access to one virtual machine within the cloud subnet. Under this assumption, the attacker's primary goal is to discover and communicate with other internal systems in order to expand control. No assumptions are made about external exploitation or credential theft.

The experiment demonstrates that, even under this limited threat model, unrestricted internal communication significantly increases risk. By applying subnet-level Network Security Group rules, the attack surface available for lateral movement is reduced, regardless of the specific technique used by the attacker.

VI. CONCLUSION

This project demonstrates that Azure Network Security Groups can successfully limit lateral movement between virtual machines when applied at the subnet level. Internal communication was unrestricted by default but fully blocked after the security rule was enforced. Connectivity tests and flow logs confirmed that containment was effective. The findings show that simple network segmentation provides meaningful protection in cloud environments and should be considered a baseline security control.The complete experimental setup, scripts and supporting evidence are available at https://github.com/63n713m4n/ Network-Worm-Containment-with-Azure-NSGs [4]

REFERENCES

[1] Palo Alto Networks Unit 42, "Unit 42 cloud threat report, volume 7," October 2023, accessed: 2025-11-17. [Online]. Available: https://www.paloaltonetworks.com/unit42/cloud-threat-report-vol-7

[2] H. A. Al-Ofeishat and R. Alshorman, "Build a secure network using segmentation and micro-segmentation techniques," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, no. 1, pp. 1048–1057, 2024. [Online]. Available: https://ijece.iaescore.com/index.php/IJECE/article/view/54671

[3] Microsoft, *What is a Network Security Group?*, Microsoft Azure Documentation, August 2023, accessed: 2025-11-17. [Online]. Available: https://learn.microsoft.com/en-us/azure/virtual-network/network-security-groups-overview

[4] group07, "Network-worm-containment-with-azure-nsgs: Supporting material for academic project," https://github.com/63n713m4n/Network-Worm-Containment-with-Azure-NSGs, January 2026, gitHub Repository. Accessed: 2026-01-15.
[5] C. Benzaïd, N. Guerd, N. El Houda Rehouma, K. Zeraoulia, and T. Taleb, "A multi-layered zero trust microsegmentation solution for cloud-native 5g & beyond networks," in *2025 IEEE Wireless Communications and Networking Conference (WCNC)*, 2025, pp. 1–7. [Online]. Available: https://doi.org/10.1109/WCNC61545.2025.10978671

VII. ACADEMIC INTEGRITY

- Brainstorming search terms: ChatGPT
- Improving grammar/clarity: ChatGPT