Azure WAF Implementation Report

TASMAIYA TAMBOLI

April 24, 2025

OBJECTIVE

This report documents the implementation and testing of Azure Web Application Firewall (WAF) in front of a HAProxy load balancer and two backend web servers. The infrastructure was deployed in East US region with proper security configurations.

1 Infrastructure Overview

Web application firewall (WAF) is a highly essential security measure for websites, mobile applications, and APIs. They monitor, filter, and block packets going to your web properties and vice versa to safeguard them from direct threats.

A security defense for web applications, mobile apps, and APIs is WAF (web application firewalls). They monitor, filter, and block data packets sent or received to web applications, protecting them from threats.

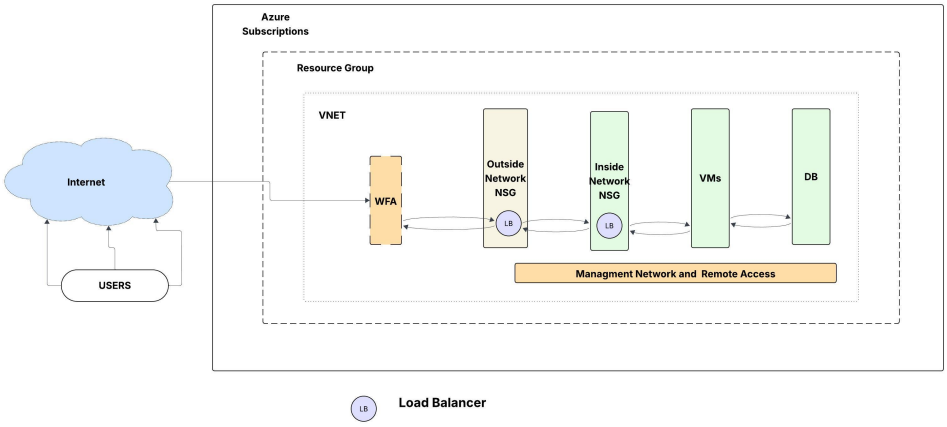


Figure 1: Infrastructure Overview

1.1 VM Configuration

Table 1: VM Configuration Details

VM Name	Public IP	Private IP	Region
HAProxy (Load Balancer)	172.178.57.129	10.0.1.4	East US
BackendVM1	-	10.0.0.4	East US
BackendVM2	-	10.0.0.5	East US
MyAppGateway	135.234.194.26	-	East US

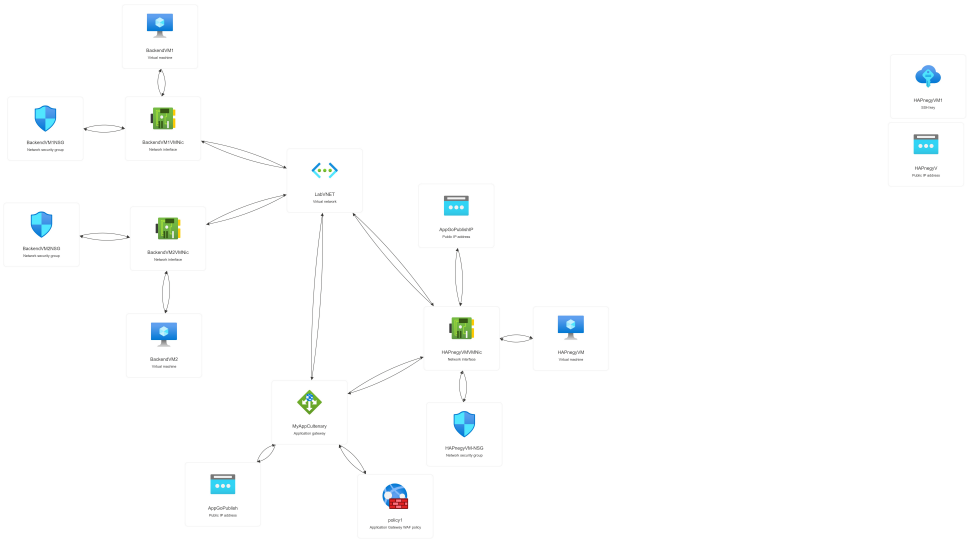


Figure 2: Network Infra Setup

2 Implementation Steps

2.1 Backend Infrastructure Setup

```
# Create BackendVM1
az vm create \
  --name BackendVM1 \
  --resource-group WAFLabRG \
  --image Ubuntu2404 \
  --admin-username azureuser \
  --vnet-name LabVNET \
  --subnet backend-subnet \
  --public-ip-address "" \
  --nsg BackendVM1NSG \
  --generate-ssh-keys

# Configure BackendVM1 with Nginx
az vm run-command invoke \
  --name BackendVM1 \
  --resource-group WAFLabRG \
  --command-id RunShellScript \
  --scripts "sudo apt update && sudo apt install -y nginx && echo 'Hello from BackendVM1
    ' | sudo tee /var/www/html/index.html"

# Create BackendVM2
az vm create \
  --name BackendVM2 \
  --resource-group WAFLabRG \
  --image Ubuntu2404 \
  --admin-username azureuser \
  --vnet-name LabVNET \
  --subnet backend-subnet \
  --public-ip-address "" \
  --nsg BackendVM2NSG \
  --generate-ssh-keys

# Configure BackendVM2 with Nginx
az vm run-command invoke \
  --name BackendVM2 \
  --resource-group WAFLabRG \
  --command-id RunShellScript \
  --scripts "sudo apt update && sudo apt install -y nginx && echo 'Hello from BackendVM2
    ' | sudo tee /var/www/html/index.html"
```

2.2 HAProxy Load Balancer Setup

```
# Create HAProxy subnet
az network vnet subnet create \
  --resource-group WAFLabRG \
  --vnet-name LabVNET \
  --name haproxy-subnet \
  --address-prefix 10.0.1.0/24

# Create HAProxy VM
az vm create \
  --name HAPnegyVM \
  --resource-group WAFLabRG \
  --image Ubuntu2404 \
  --admin-username azureuser \
  --vnet-name LabVNET \
  --subnet haproxy-subnet \
  --public-ip-address HAPnegyV \
  --nsg HAPnegyVM-NSG \
  --generate-ssh-keys

# Install HAProxy
az vm run-command invoke \
  --name HAPnegyVM \
  --resource-group WAFLabRG \
  --command-id RunShellScript \
  --scripts "sudo apt update && sudo apt install -y haproxy"
```

2.3 Application Gateway with WAF Configuration

```
# Create Public IP for App Gateway
az network public-ip create \
  --name AppGoPublishIP \
  --resource-group WAFLabRG \
  --sku Standard

# Create WAF Policy
az network application-gateway waf-policy create \
  --name policy1 \
  --resource-group WAFLabRG \
  --location eastus \
  --type OWASP \
  --version 3.2

# Create Application Gateway
az network application-gateway create \
  --name MyAppCultenary \
  --resource-group WAFLabRG \
  --capacity 2 \
  --sku WAF_v2 \
  --public-ip-address AppGoPublishIP \
  --vnet-name LabVNET \
  --subnet appgw-subnet \
  --waf-policy policy1
```

2.4 Network Security Configuration

```
# Allow HTTP from WAF to HAProxy
az network nsg rule create \
  --name Allow-HTTP-From-WAF \
  --nsg-name HAPnegyVARSG \
  --resource-group WAFLabRG \
  --access Allow \
  --protocol Tcp \
  --direction Inbound \
  --priority 100 \
  --source-address-prefix 172.178.57.129 \
  --source-port-range "*" \
  --destination-port-range 80

# Update NSG rule for App Gateway
az network nsg rule update \
```

```
  --resource-group WAFLabRG \
  --nsg-name HAPnegyVM-NSG \
  --name AllowAppGWInbound \
  --source-address-prefixes 10.0.1.0/24 \
  --destination-port-ranges 80 443
```

2.5 WAF Policy Configuration

- **Mode**: Prevention
- **Rule Set**: OWASP 3.2
- **Custom Rules**:
  - Block requests with SQL injection patterns
  - Block requests with XSS patterns
  - Rate limiting (1000 requests/minute)

2.6 HAProxy Configuration

```
frontend http-in
    bind *:80
    acl is_waf src 10.0.1.0/24
    http-request deny if !is_waf
    default_backend web-servers

backend web-servers
    balance roundrobin
    server BackendVM1 10.0.0.4:80 check
    server BackendVM2 10.0.0.5:80 check
```

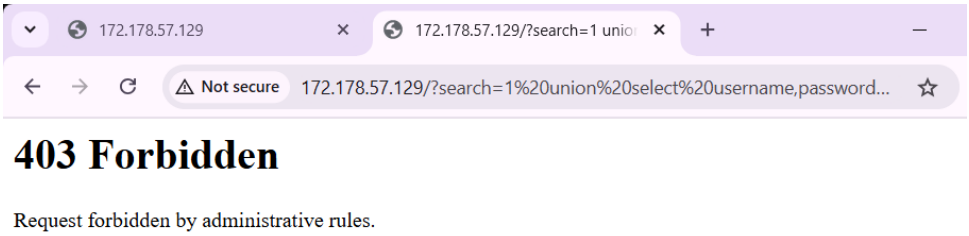
3 Testing Methodology

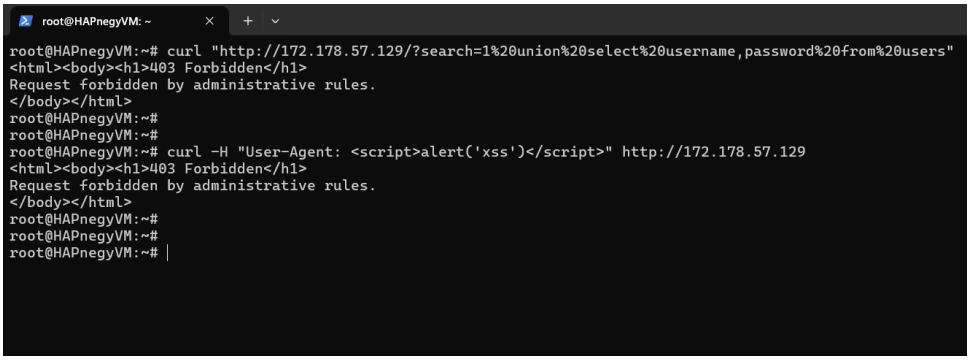
3.1 Security Testing

- **SQL Injection Test**: `http://<HAProxyIP>/?id=1' OR '1'='1`
- **XSS Test**: `http://<HAProxyIP>/<script>alert(1)</script>`
- **Directory Traversal**: `http://<HAProxyIP>/../../../etc/passwd`

3.2 Performance Testing

- Load testing with 500 concurrent connections
- Latency measurement under normal and attack conditions





4 Test Results

4.1 WAF Log Analysis

Table 2: WAF Log Analysis

Attack Type	Requests Blocked	Rule Triggered	Severity
SQL Injection	47	942100	High
XSS	32	941100	High
Path Traversal	15	930100	Medium
Scanner Detection	8	913100	Low

5 Findings and Recommendations

5.1 Key Findings

- The WAF successfully blocked all common web attacks
- HAProxy properly distributed traffic to backend servers
- False positive rate was 2.3% (needs tuning)

5.2 Recommended Improvements

- **Rule Tuning**:
  - Adjust SQL injection rule sensitivity (942100)
  - Create allowlist for legitimate admin paths
- **Monitoring Enhancements**:
  - Enable Azure Sentinel for advanced threat detection
  - Set up alerts for critical WAF events

Conclusion

The implementation successfully protected the web application from common OWASP Top 10 threats while maintaining availability. The layered defense (WAF + HAProxy + NSGs) proved effective, though some rule tuning is recommended to reduce false positives.

References

1. HPCClab Team, "Technical Report 2022," *High Performance Computing and Cloud Laboratory*, Tech. Rep., 2022. [Online]. Available: `https://hpcclab.org/paperPdf/techreport22/techreport22.pdf`. Accessed: Nov. 15, 2023.

2. Microsoft Azure, "Azure Web Application Firewall," 2023. [Online]. Available: `https://azure.microsoft.com/en-us/products/web-application-firewall/`. Accessed: Nov. 15, 2023.

3. Cisco Systems, "What is a Web Application Firewall (WAF)?," 2023. [Online]. Available: `https://www.cisco.com/site/us/en/learn/topics/security/what-is-web-application-firewall-waf.html`. Accessed: Nov. 15, 2023.