Best Security Practices with Deploying Virtual Machines in mCloud

Micron21 Support - 2025-04-04 - Security

Just like you would lock your doors and windows to protect your home, this process involves updating software, disabling unnecessary services, and tightening access controls.
This article will briefly explore these steps to help keep your server safe and reliable. This is not a comprehensive list, but makes a great starting point for touching base for the most common security pitfalls.

Minimise Running Services and Installed Software

- Remove unnecessary packages and services from operating systems or container images.
- Prevent the installation of software or services not required for the purposes of your server.

  The aim is to reduce the attack surface, and make it easier to keep track of what’s meant to be running.

Keep Systems Patched

- Apply updates and security patches regularly to stay protected against known vulnerabilities.
- Ensure updates and security patches come from trusted sources, and are cryptographically signed
- Schedule patch cycles or use automated tools to keep everything current.

  The aim is to prevent newly discovered vulnerabilities from being exploited on your server.

Apply Principles of Least-Privilege

- Ensure that each application and user has only the minimum permissions required to carry out it’s task.
- Limit sudo access and adopt role-based access control (RBAC) if possible.
- Restrict read and write permissions to files and databases that may contain sensitive

data

The aim is to minimise the risk of an application or user which becomes compromised from allowing an attacker to move through your server or network.

Enable Logging and Monitoring

- Turn on logging and auditing for system and application events. If possible, ensure logging is immutable or cannot otherwise be tampered with.
- Use monitoring tools (e.g., ELK Stack, Prometheus) configured to generate alerts on suspicious activity.

  The aim is to be aware of any problems before, or as they are starting to unfold, rather than after the fact.

Lock Down Your Network, and Network Ports

- By default, block all inbound connections and only open the ports that are strictly needed.
- If possible, by default, block all outgoing connections, and only open the ports that are strictly needed.
- If possible, ensure every firewall allow-rule contains a specific Source, Destination, Port, and Protocol, rather than “any” or “all”.
- Implement network segmentation using sub-netting or VLAN segments so that network traffic is tightly controlled

  The aim is to prevent lateral movement throughout the network, and to reduce the attack surface of your server.

Perform Routine Network Scans or Penetration Testing

- Server hardening is not “set and forget”; having a routine network scan or scheduled penetration testing is recommended

  The aim is to prevent security holes from opening up, or going unnoticed as configuration changes or updates take place.

Deploy Network and Host Intrusion Detection/Prevention

- Tools like Snort, Suricata, fail2ban, or some commercial offerings can detect and respond to abnormal traffic or repeated failed login attempts.
- At the firewall or edge router level, use advanced filtering and network and application rate limits where possible.

  The aim is to empower systems and networks to self-isolate and autonomously lock down in the event that a malicious attack is detected, and a quick response is needed.

Use Strong Encryption and Authentication

- Enforce SSH key-based login instead of passwords. Regular key rotation is also recommended.
- Use strong and secure multi-factor authentication (MFA) methods where possible.
- Encrypt data in transit (i.e. with HTTPS/TLS) and do not re-use private keys for new certificates.
- Encrypt sensitive data at rest if feasible.

  The aim is to minimise the risk of a successful brute force attack being successful on your network data stream or authentication endpoints.

Have a Backup Plan and a Disaster Recovery Plan

- Have a backup schedule that suits your level of risk; e.g. nightly backups, with 14 days retention.
- Have a plan on how and when to use your backups to recover from disaster
- If possible, have backups completely separate from your network. Better yet, store backups on immutable storage.
- Test that your backups, and that your disaster recovery plan, actually work - and know how long it will take to carry out.

  The aim is to mitigate the potential downtime from a successful attack, and to prevent having to pay a malicious actor who holds your network, server, or data to ransom.