Capitolo 9 – Hardening e Sicurezza dei Server

Dopo aver imparato a deployare applicazioni, è fondamentale proteggere i server.

L'**hardening** consiste nell'applicare configurazioni e best practice per ridurre la superficie di attacco e aumentare la sicurezza.

Ansible è uno strumento ideale per automatizzare queste procedure.

9.1 Aggiornamenti di Sicurezza

Eseguire regolarmente aggiornamenti di sicurezza è fondamentale.

Esempio playbook:

```
- name: Aggiornamenti di sicurezza
  hosts: all
  become: true
  tasks:
    - name: Aggiornamento pacchetti Debian
      apt:
        upgrade: dist
      when: ansible_os_family == "Debian"

    - name: Aggiornamento pacchetti RedHat
      yum:
        name: "*"
        state: latest
      when: ansible_os_family == "RedHat"
```

9.2 Gestione Utenti e SSH

- Disabilitare login root diretto via SSH
- Creare utenti con privilegi limitati
- Forzare l’uso di chiavi SSH
- Disabilitare password login quando possibile

Esempio task per SSH:

```
- name: Disabilita root login SSH
  lineinfile:
    path: /etc/ssh/sshd_config
    regexp: '^PermitRootLogin'
    line: 'PermitRootLogin no'
  notify: Restart ssh

- name: Imposta chiavi SSH per utente deploy
  authorized_key:
    user: deploy
    state: present
    key: "{{ lookup('file', '~/.ssh/id_ed25519.pub') }}"
```

Handler per restart SSH:

```
- name: Restart ssh
  service:
    name: ssh
    state: restarted
```

9.3 Firewall e Accesso

Abilitare firewall con regole base:

```
- name: Configura UFW
  ufw:
    state: enabled
    rule: allow
```

```
    port: "{{ item }}"
  loop:
    - 22
    - 80
    - 443
```

- Port 22 → SSH
- Port 80 → HTTP
- Port 443 → HTTPS

9.4 Rimozione Pacchetti Non Necessari

Ridurre il rischio rimuovendo software inutile:

```
- name: Rimuovi pacchetti non necessari
  package:
    name: "{{ item }}"
    state: absent
  loop:
    - telnet
    - ftp
    - rsh-client
```

9.5 Logging e Audit

Abilitare logging e audit per monitorare accessi:

```
- name: Installazione auditd
  package:
    name: auditd
    state: present

- name: Avvia auditd
```

```
  service:
    name: auditd
    state: started
    enabled: yes
```

9.6 Sicurezza dei File di Configurazione

- Impostare permessi restrittivi su file sensibili
- Separare variabili segrete con Ansible Vault

Esempio:

```
- name: Imposta permessi su segreti
  file:
    path: /etc/app/secrets.yml
    owner: root
    group: root
    mode: '0600'
```

9.7 Best Practice Hardening

- Automatizzare tutto con Ansible
- Versionare playbook di sicurezza separatamente
- Testare ogni modifica in staging prima di prod
- Monitorare log di sistema e anomalie
- Aggiornare regolarmente le policy di sicurezza

9.8 Conclusione

Applicare hardening con Ansible permette di:

- Ridurre la superficie di attacco

- Automatizzare best practice di sicurezza
- Mantenere server coerenti e sicuri
- Avere procedure ripetibili e verificabili

Nei prossimi capitoli si può approfondire:

- Logging e monitoring avanzato
- Troubleshooting playbook

Revision #2
Created 2026-02-26 14:11:27 UTC by Pe
Updated 2026-02-26 14:14:06 UTC by Pe