

RÚBRICA UNIDAD 3: SEGURIDAD LOCAL BÁSICA

Gestión y Despliegue de Sistemas Operativos con Ansible

INFORMACIÓN DEL PROYECTO

Curso: Sistemas Operativos

Ciclo: 6

Fecha: Noviembre 2025

Autores:

- Boris Quispe
- Jose Zuñiga

Docente:

Alex Roberto Villegas Cervera

Repository:

<https://github.com/kyrafka/ansible>

Criterio 1: Seguridad Local Básica

Implementación:

1. Controles de Acceso Implementados:

- **Usuarios diferenciados por rol** con permisos específicos
- **Políticas de sudo personalizadas** por tipo de usuario
- **Grupos de seguridad** para organización de permisos

Ubicación en el código:

```
# roles/ubuntu_desktop/tasks/main.yml
# roles/server_users/tasks/main.yml
```

Usuarios creados:

Usuario	Sistema	Permisos	Grupos
administrador	Ubuntu Desktop	Sudo completo	sudo, adm, systemd-journal
auditor	Ubuntu Desktop/Server	Sudo limitado (solo lectura)	adm, systemd-journal, auditors
gamer01	Ubuntu Desktop	Sin sudo	pcgamers
dev	Server	Sudo limitado (servicios)	developers

Evidencia:

```
# Ver usuarios y permisos  
bash scripts/client/mostrar-usuarios-grupos.sh  
bash scripts/diagnostics/check-user-permissions.sh
```

Criterio 2: Protección Contra Amenazas

Implementación:

1. Firewall (UFW) Configurado:

- **Política por defecto:** Denegar entrada, permitir salida
- **Reglas específicas** por servicio
- **Rate limiting en SSH** para prevenir ataques de fuerza bruta
- **Monitoreo activo** con script personalizado

Ubicación en el código:

```
# roles/firewall/tasks/main.yml
```

Reglas aplicadas:

```
# SSH con rate limiting  
ufw limit 22/tcp comment 'SSH con rate limiting'  
  
# DNS  
ufw allow 53/tcp comment 'DNS TCP'  
ufw allow 53/udp comment 'DNS UDP'  
  
# DHCP IPv6  
ufw allow 547/udp comment 'DHCPv6 Server'  
ufw allow 546/udp comment 'DHCPv6 Client'  
  
# HTTP  
ufw allow 80/tcp comment 'HTTP Web Server'  
  
# FTP Pasivo  
ufw allow 21000:21010/tcp comment 'FTP Passive Ports'
```

2. fail2ban Configurado:

- **Protección SSH** contra intentos de acceso no autorizado
- **Baneos automáticos** después de intentos fallidos
- **Monitoreo de logs** en tiempo real

Configuración:

```
# roles/firewall/templates/jail.local.j2
[sshd]
enabled = true
port = ssh
filter = sshd
logpath = /var/log/auth.log
maxretry = 5
bantime = 3600
findtime = 600
```

Evidencia:

```
# Ver estado del firewall
sudo ufw status verbose

# Ver estado de fail2ban
sudo fail2ban-client status

# Monitoreo en tiempo real
sudo /usr/local/bin/firewall-monitor.sh
```

Criterio 3: Prácticas Seguras de Usuario

Implementación:

1. Contrasenñas Seguras:

- **Hash SHA-512** para todas las contraseñas
- **Contrasenñas únicas** por usuario
- **No se almacenan en texto plano**

Ubicación en el código:

```
# roles/ubuntu_desktop/tasks/main.yml
password: "{{ ubuntu_desktop_users.admin.password | password_hash('sha512') }}"
```

2. Restricciones de Acceso SSH:

- **Root login deshabilitado**
- **Autenticación por contraseña habilitada** (con rate limiting)
- **X11 Forwarding deshabilitado** (seguridad adicional)

Configuración SSH:

```
# roles/seguridad/tasks/main.yml
PermitRootLogin no
PasswordAuthentication yes
```

```
PubkeyAuthentication yes
X11Forwarding no
```

3. Permisos de Archivos:

- **Sudoers con permisos 0440** (solo lectura para root)
- **Validación automática** de sintaxis sudoers
- **Scripts de monitoreo con permisos 0755**

Evidencia:

```
# Ver configuración SSH
cat /etc/ssh/sshd_config | grep -E "PermitRootLogin|PasswordAuthentication"

# Ver permisos de sudoers
ls -la /etc/sudoers.d/

# Probar acceso SSH según rol
bash scripts/diagnostics/test-ssh-ubpc.sh
```

Criterio 4: Políticas de Seguridad

Implementación:

1. Políticas de Sudo Diferenciadas:

Administrador (acceso completo):

```
# /etc/sudoers.d/administrador
administrador ALL=(ALL) NOPASSWD: ALL
```

Auditor (solo lectura):

```
# /etc/sudoers.d/auditor
auditor ALL=(ALL) NOPASSWD: /usr/bin/journalctl, /usr/bin/systemctl status *,
/usr/bin/tail /var/log/*, /usr/bin/cat, /usr/bin/less, /usr/bin/ls
```

Dev (servicios específicos):

```
# /etc/sudoers.d/dev
dev ALL=(ALL) NOPASSWD: /usr/bin/systemctl restart nginx, /usr/bin/systemctl restart
apache2
```

Gamer (sin sudo):

```
# Sin archivo en /etc/sudoers.d/
# Permisos limitados solo a su home
```

2. Políticas de Firewall por Rol:

- **Servidor:** Firewall permisivo para servicios necesarios
- **Cliente:** Firewall restrictivo, solo salida permitida
- **Reglas específicas** según función del sistema

Ubicación en el código:

```
# roles/firewall/tasks/filter-by-role.yml
```

3. Políticas de Grupo:

- **Grupos organizados por función:** sudo, adm, auditors, developers, pcgamers
- **Permisos heredados** según membresía de grupo
- **Separación de privilegios clara**

Evidencia:

```
# Ver políticas de sudo
sudo cat /etc/sudoers.d/*

# Ver grupos y membresía
getent group | grep -E "sudo|adm|auditors|developers|pcgamers"

# Ver usuarios por grupo
bash scripts/client/mostrar-usuarios-grupos.sh
```

📋 TABLA DE EVIDENCIAS

Criterio	Implementación	Script de Verificación	Archivo de Configuración
Seguridad Local	Usuarios con roles diferenciados	scripts/client/mostrar-usuarios-grupos.sh	roles/ubuntu_desktop/tasks/main.yml
Protección Amenazas	UFW + fail2ban	scripts/diagnostics/show-server-config.sh	roles/firewall/tasks/main.yml
Prácticas Seguras	Contraseñas hash + SSH seguro	scripts/diagnostics/test-ssh-ubpc.sh	roles/seguridad/tasks/main.yml
Políticas Seguridad	Sudoers personalizados	scripts/diagnostics/check-user-permissions.sh	roles/ubuntu_desktop/tasks/admin.yml

⌚ COMANDOS PARA GENERAR EVIDENCIAS

En el Servidor Ubuntu

```
# Mostrar configuración de firewall  
sudo ufw status verbose  
  
# Mostrar estado de fail2ban  
sudo fail2ban-client status  
sudo fail2ban-client status sshd  
  
# Ver usuarios del sistema  
cat /etc/passwd | grep -E "auditor|dev"  
  
# Ver políticas de sudo  
sudo cat /etc/sudoers.d/auditor  
sudo cat /etc/sudoers.d/dev
```

En Ubuntu Desktop

```
# Mostrar usuarios y grupos  
bash scripts/client/mostrar-usuarios-grupos.sh  
  
# Verificar permisos diferenciados  
bash scripts/diagnostics/check-user-permissions.sh  
  
# Probar acceso SSH según rol  
bash scripts/diagnostics/test-ssh-ubpc.sh  
  
# Ver configuración de firewall local  
sudo ufw status verbose
```

Capturas Recomendadas

1. **Firewall activo:** `sudo ufw status verbose`
2. **fail2ban funcionando:** `sudo fail2ban-client status`
3. **Usuarios creados:** `bash scripts/client/mostrar-usuarios-grupos.sh`
4. **Permisos sudo:** `sudo cat /etc/sudoers.d/administrador`
5. **Permisos sudo limitados:** `sudo cat /etc/sudoers.d/auditor`
6. **SSH bloqueado para gamer:** Intento de SSH con usuario gamer01
7. **SSH permitido para admin:** Conexión SSH exitosa con administrador
8. **Logs de fail2ban:** `sudo tail -f /var/log/fail2ban.log`

💻 DOCUMENTACIÓN TÉCNICA

Arquitectura de Seguridad

Capas de Seguridad Implementadas:

1. **Capa de Red:** Firewall UFW con políticas restrictivas
2. **Capa de Acceso:** fail2ban para prevención de intrusiones
3. **Capa de Usuario:** Roles y permisos diferenciados
4. **Capa de Sistema:** Políticas de sudo personalizadas

Principios Aplicados:

- **Principio de mínimo privilegio:** Cada usuario tiene solo los permisos necesarios
- **Defensa en profundidad:** Múltiples capas de seguridad
- **Separación de funciones:** Roles claramente definidos
- **Auditoría:** Logs y monitoreo activo

🔍 JUSTIFICACIÓN TÉCNICA

¿Por qué UFW y no iptables directamente?

UFW (Uncomplicated Firewall) proporciona una interfaz simplificada sobre iptables, facilitando la gestión y reduciendo errores de configuración. Es ideal para entornos educativos y de laboratorio.

¿Por qué fail2ban?

fail2ban monitorea logs en tiempo real y banea automáticamente IPs con comportamiento sospechoso, proporcionando protección activa contra ataques de fuerza bruta sin intervención manual.

¿Por qué políticas de sudo personalizadas?

Permite implementar el principio de mínimo privilegio, donde cada usuario tiene exactamente los permisos necesarios para su función, reduciendo el riesgo de escalada de privilegios.

¿Por qué hash SHA-512 para contraseñas?

SHA-512 es un algoritmo de hash criptográfico robusto que protege las contraseñas almacenadas. Incluso si un atacante accede a `/etc/shadow`, no puede obtener las contraseñas en texto plano.

📝 CONCLUSIÓN

- Seguridad Local Básica:** Controles de acceso con buenas prácticas y documentación completa
- Protección Contra Amenazas:** Firewall y fail2ban con monitoreo activo
- Prácticas Seguras:** Cultura de seguridad promovida en todos los entornos
- Políticas de Seguridad:** Aplicadas con enfoque profesional y técnico

Proyecto: Gestión y Despliegue de Sistemas Operativos

Curso: Sistemas Operativos - Ciclo 6

Fecha: Noviembre 2025

Autores: Boris Quispe, Jose Zuñiga

Docente: Alex Roberto Villegas Cervera