



# Seguridad



# Infraestructura



#SOMOSVALLIANS



00

# Quién Soy?

accenture



# José Antonio Garcia Cartagena

Solutions Architect @AWS

01

# Fundamentos



## ¿Por qué es importante?

- 43% de ciberataques apuntan a pequeñas empresas
- Coste medio de una brecha: **\$4.45M (2023)**
- Tiempo medio para detectar: **287 días**

### Consecuencias:

- Pérdida de datos sensibles
  - Daño reputacional
  - Multas regulatorias (GDPR)
  - Interrupción del servicio
- 
- <https://www.salford.ac.uk/business/greater-manchester-cyber-foundry/cybersecurity-isnt-a-priority-for-smes-right-change-your-strategy#:~:text=Cyber%20security%20statistics%20show%20that%2043%25%20of,of%20a%20cyber%2Dattack%20go%20out%20of%20business>
  - <https://www.brontobytecloud.com/la-ciberdelincuencia-es-costosa-pero-cuanto-cuesta-realmente#:~:text=El%20coste%20medio%20de%20una,sin%20contar%20el%20da%C3%B1o%20reputacional>.
  - <https://blog.camelsecure.com/287-d%C3%ADas-promedio-tarda-identificar-y-contener-brechas>

# Los 3 Pilares de la Seguridad (CIA Triad)

Modelo fundamental de seguridad de la información que guía las políticas de ciberseguridad

## • CONFIDENCIALIDAD

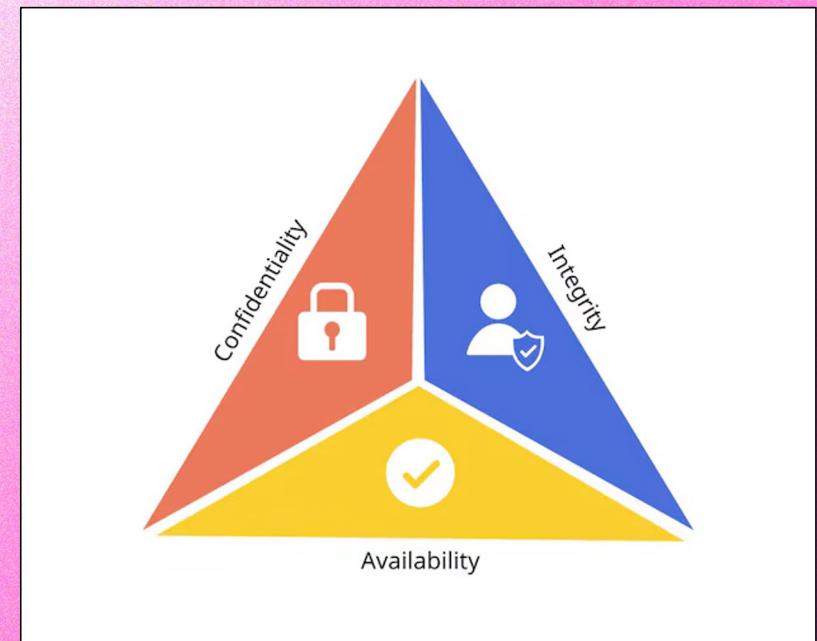
- Garantiza solo acceso autorizado a la información
- *Cifrado de contraseñas en BD*

## • INTEGRIDAD

- Protege que esten los datos correctos y no modificados
- *Checksums en descargas*

## • DISPONIBILIDAD

- Asegura el acceso cuando se necesita
- *Backups y redundancia*



# Principio de Mínimo Privilegio

*"Dar solo los permisos necesarios, nada más"*

 **MAL: Usuario con acceso root innecesario**

user: root  
permissions: ALL

 **BIEN: Usuario con permisos específicos**

user: app\_user  
permissions: read /app/data, write /app/logs

## Beneficios:

- Limita el daño en caso de compromiso
- Reduce superficie de ataque
- Facilita auditorías



## Defensa en Profundidad

*"Múltiples capas de seguridad"*

[Firewall] ← Capa 1: Perímetro

[WAF] ← Capa 2: Aplicación

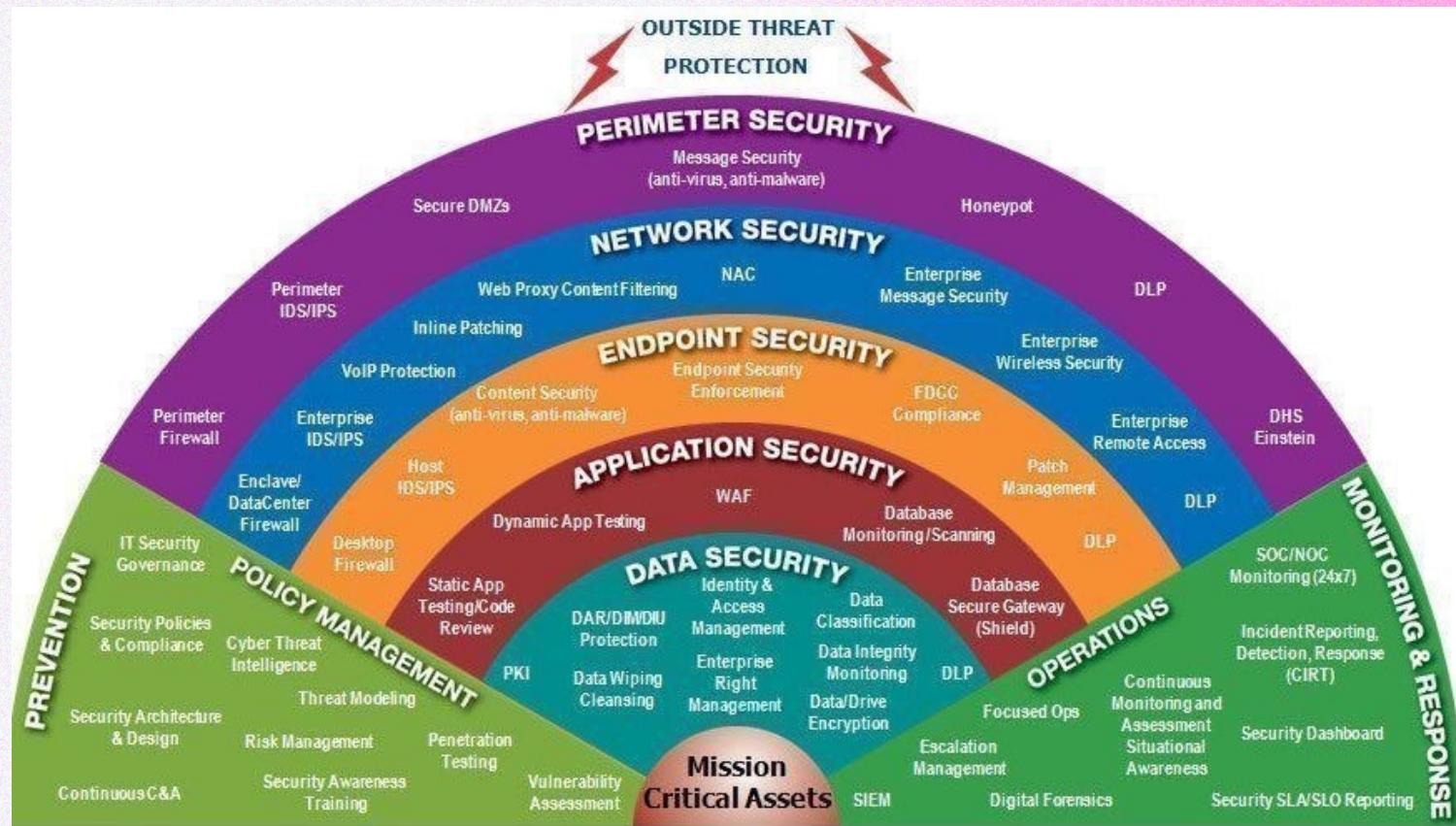
[Autenticación] ← Capa 3: Identidad

[Cifrado] ← Capa 4: Datos

[Logs/Monitoreo] ← Capa 5: Detección



# Defensa en Profundidad



# Vectores de Ataque Comunes

## Exposición de Servicios:

- Puertos abiertos innecesariamente
- Servicios sin autenticación

## Configuraciones Inseguras

- Permisos excesivos
- Cifrado deshabilitado

## Credenciales Débiles

- Contraseñas por defecto
- Secretos en código

## Falta de Actualizaciones

- Software desactualizado
- Vulnerabilidades conocidas (CVEs)

## ✗ NUNCA hagas esto:

```
DB_PASSWORD = "admin123"  
API_KEY = "sk-1234567890abcdef"
```

## ✓ Usa gestores de secretos:

```
import os  
DB_PASSWORD = os.getenv('DB_PASSWORD')  
API_KEY = os.getenv('API_KEY')
```

### Herramientas:

- Variables de entorno
- HashiCorp Vault
- AWS Secrets Manager
- Docker Secrets



# Seguridad en Contenedores

## Mejores prácticas:

- **Imágenes base oficiales y actualizadas**  
FROM python:3.11-slim
- **Escaneo de vulnerabilidades**  
docker scan myimage:latest
- **Usuario no-root**  
USER appuser
- **Límites de recursos**  
limits: cpus: '0.5', memory: 512M

## 1. Segmentación de red

- Redes separadas por función
- DMZ para servicios públicos

## 2. Firewall rules

- allow: 443/tcp (HTTPS)
- allow: 22/tcp from 10.0.0.0/8 (SSH interno)
- deny: all (Denegar todo lo demás)

## 3. Cifrado en tránsito

- TLS/SSL para comunicaciones
- VPN para acceso remoto



# Logging y Monitoreo

*"No puedes proteger lo que no puedes ver"*

## Que registrar

- Intentos de autenticación
- Cambios de configuración
- Accesos a datos sensibles
- Errores y excepciones
- Tráfico de red anómalo

## Herramientas

- ELK Stack (Elasticsearch, Logstash, Kibana)
- Prometheus + Grafana
- CloudWatch (AWS)



## Checklist de Seguridad Básica

- Principio de mínimo privilegio aplicado
- Secretos fuera del código
- Cifrado en reposo y en tránsito
- Autenticación fuerte (MFA cuando sea posible)
- Firewall configurado correctamente
- Logs habilitados y monitoreados
- Backups regulares y probados
- Software actualizado
- Escaneo de vulnerabilidades regular
- Plan de respuesta a incidentes

02

## SPOT THE RISK



## SPOT THE RISK: Encuentra las Vulnerabilidades

**Objetivo: Identificar todos los problemas de seguridad**

Puntos por severidad:

- Crítico: 5 puntos
- Alto: 3 puntos
- Medio: 2 puntos

*Pista: Hay 15 vulnerabilidades escondidas*



## Solución – Vulnerabilidades Críticas

### ● CRÍTICAS (5 pts c/u):

1. Firewall completamente abierto (0.0.0.0/0) → *Exposición total de servicios internos*
2. Secretos hardcodeados en código → *Credenciales en repositorios Git*
3. Base de datos PostgreSQL expuesta públicamente → *Acceso directo a datos sensibles*
4. Redis sin autenticación → *Lectura/escritura sin control*
5. Credenciales por defecto (admin/admin) → *Acceso administrativo trivial*

### ● ALTAS (3 pts c/u):

- 6. HTTP sin cifrar 7. Servicios como root
- 8. PostgreSQL 9.6 desactualizado 9. SSH con password público
- 10. Backups sin cifrar en /tmp

### ● MEDIAS (2 pts c/u):

- 11. Logs deshabilitados 12. Sin segmentación de red
- 13. Sin límites de recursos 14. Admin panel público
- 15. Sin monitoreo ni alertas

03

# Resumen



## Top 10 Mejores Prácticas

1. NUNCA expongas servicios sin autenticación
2. SIEMPRE usa gestores de secretos
3. EVITA ejecutar como root
4. CIFRA datos sensibles (reposo + tránsito)
5. CONFIGURA firewalls restrictivos
6. HABILITA logging y monitoreo
7. ACTUALIZA software regularmente
8. ESCANEA vulnerabilidades periódicamente
9. REALIZA backups y pruébalos
10. DOCUMENTA tu arquitectura de seguridad



# ¡Kahoot Time!





# ¡GRACIAS!



Jose Antonio Garcia Cartagena



Jantcart@amazon.es



thevallians.com

v