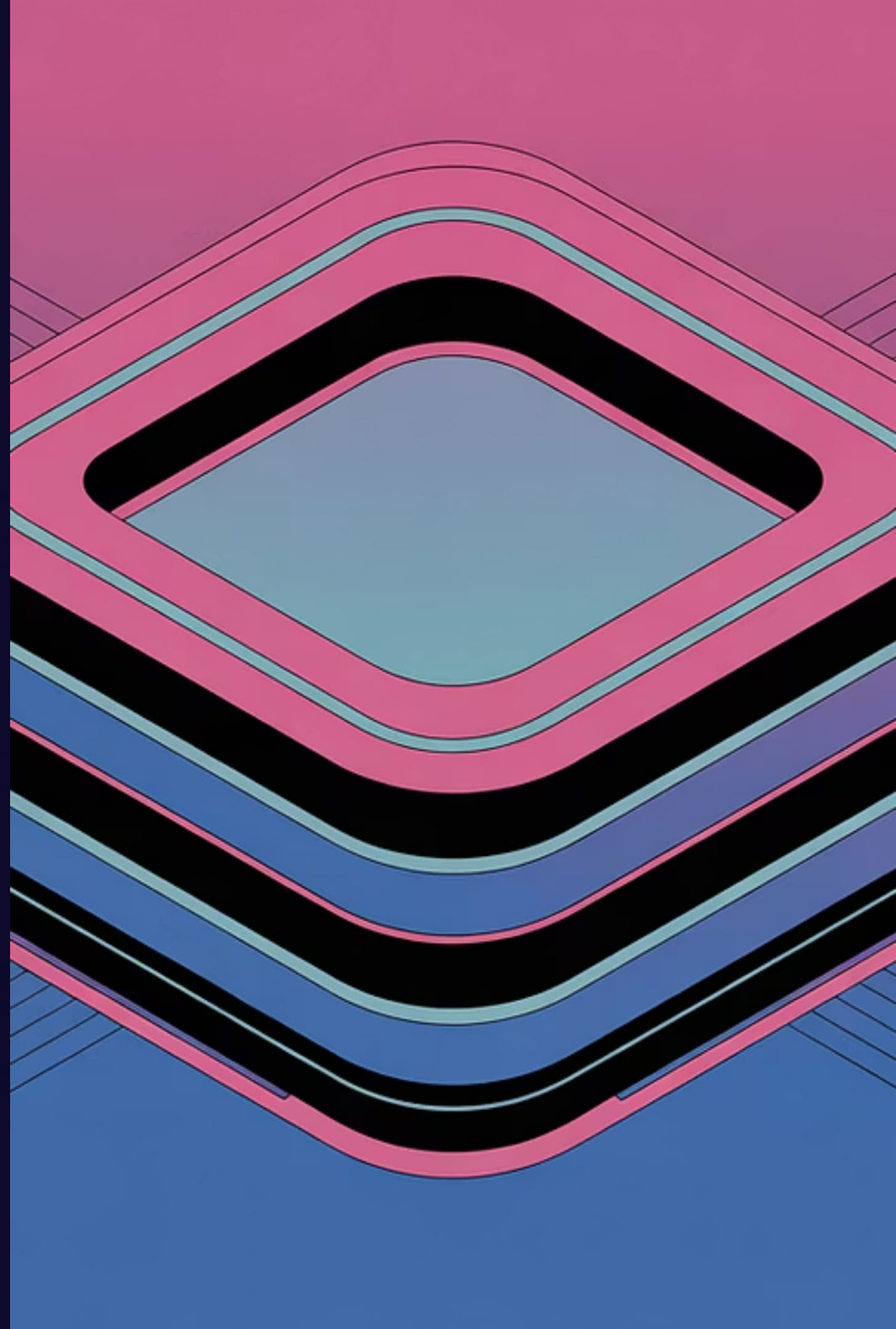
Secure Architecture Principles and Practices

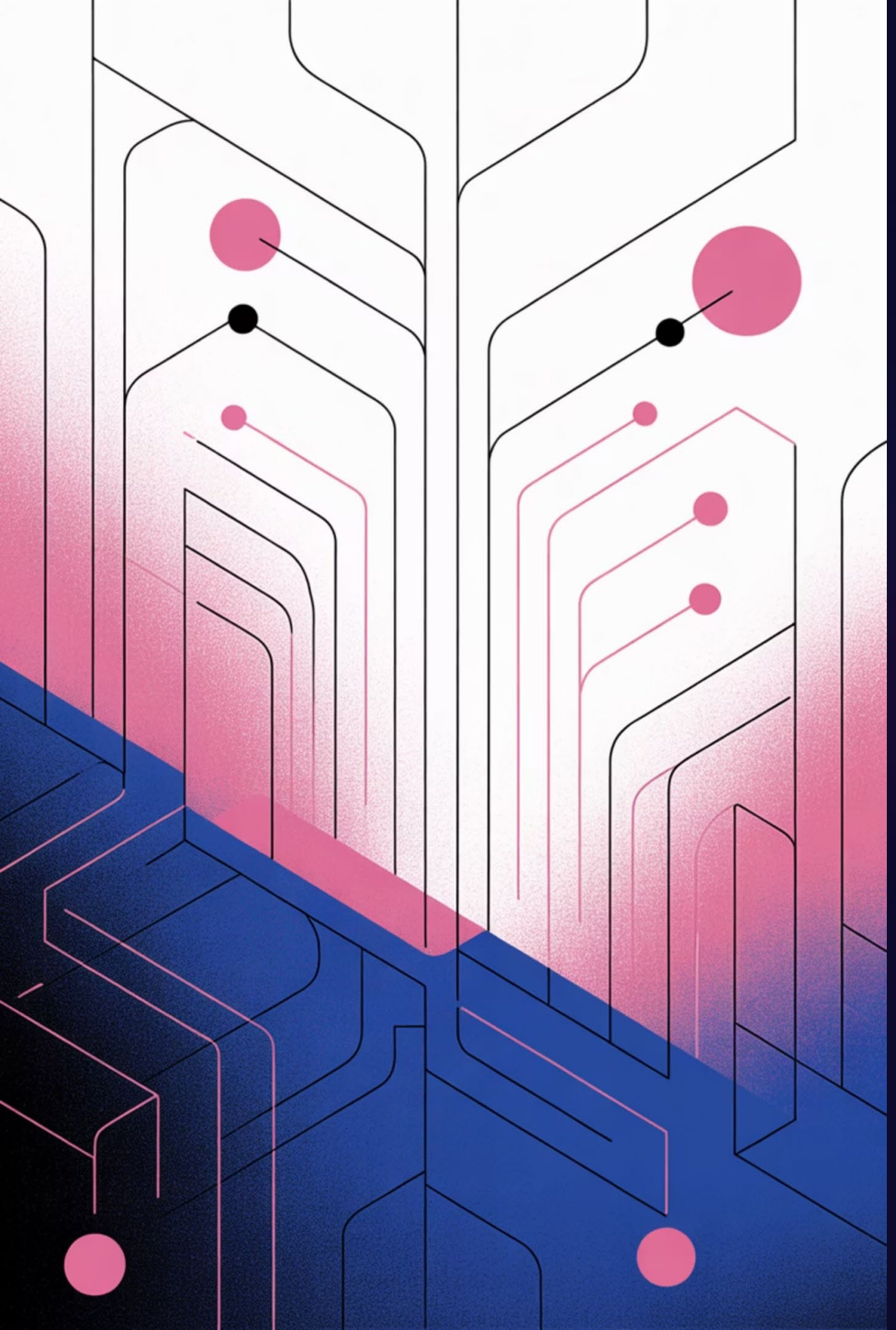
Comprehensive Guide to Practices 9-11 and Integration



Practice 9

Enforce Least Functionality and Secure Configurations

What It Means: Only enable the specific programs, services, ports, protocols, and functions you actually need. Disable, restrict, or completely remove everything else. Every enabled feature is a potential attack vector.



The Attack Surface Concept

Your attack surface is the sum of all points where an attacker could try to breach your system. The larger your attack surface, the more opportunities attackers have. Least functionality minimises this surface area.

How to Implement Least Functionality

01

Conduct a Comprehensive Audit

- Inventory all running services and programmes
- Document all open network ports
- List all enabled protocols
- Identify all installed applications
- Review all enabled system functions

02

Identify Essential Functions

- Determine what's actually needed for business operations
- Distinguish between "required" and "nice to have"
- Consult with users and stakeholders
- Document the business justification for each service

03

Disable Unnecessary Services

- Turn off services not required for operations
- Stop services that only run occasionally (enable them only when needed)
- Remove services that have secure alternatives

Implementation Steps Continued

1

Remove Nonessential Programmes

- Uninstall applications you don't actively use
- Remove trial software and bloatware
- Eliminate redundant tools
- Get rid of outdated or deprecated software

2

Close Unused Ports and Block Unneeded Protocols

- Configure firewalls to block ports you're not using
- Disable network protocols not required (e.g., if you don't need IPv6, disable it)
- Use whitelisting rather than blacklisting when possible

Secure Administrative Access

Critical Rule: Never manage systems over unencrypted protocols

- Use SSH instead of Telnet for remote command line access
- Use HTTPS instead of HTTP for web-based administration
- Implement jump servers or bastion hosts for administrative access
- Require multi-factor authentication for administrative interfaces

Documentation and Management

Step 7: Document Your Baseline Configuration

- Create detailed documentation of your approved configuration
- Maintain a configuration management database
- Version control your configuration files
- Create templates for consistent deployment

Step 8: Implement Configuration Management

- Use tools to enforce your baseline configuration
- Automatically detect configuration drift
- Restore proper configuration when changes are detected
- Track all configuration changes with full audit trails

The Bottom Line

1	2	3
If you don't need it	If you can't remove it	If you can't disable it
Remove it	**Disable it**	**Restrict it**

Every unnecessary component is a liability.

Practice 10

Manage Default Accounts and Settings

What It Means: Never use the default configurations that come with new systems, applications, or devices. Change or disable all default administrative accounts and passwords. Create and enforce a documented secure configuration standard across your entire environment.

Why Defaults Are Dangerous

Manufacturers and developers use default settings to make installation easy, not secure. These defaults are:

- Publicly documented
- Identical across thousands or millions of devices
- Indexed in attacker databases
- Targeted by automated scanning tools
- The first thing attackers try

How to Implement Default Account Management

Step 1: Create a Secure Configuration Standard

- Document approved settings for each type of system
- Include security-hardening steps in your standard
- Base your standard on industry benchmarks (CIS Benchmarks, DISA STIGs, vendor hardening guides)
- Get stakeholder approval for the standard
- Keep the standard updated as threats and technology evolve

Step 2: Address Default Accounts

For accounts you can disable:

- Disable built-in accounts like "administrator," "root," "admin," "guest"
- Create new administrative accounts with unique, non-obvious names
- Document which accounts are disabled and why

For accounts you cannot disable:

- Change the password to something extremely strong (long, random, complex)
- Set account policies that make the account unusable (e.g., deny login rights)
- Limit where the account can log in from
- Monitor the account closely for any usage attempts

Common Default Accounts to Address

Windows
Administrator, Guest, DefaultAccount

Linux
root, admin, user

Databases
sa (SQL Server), root (MySQL), postgres (PostgreSQL), SYSTEM (Oracle)

Network devices
admin, cisco, ubnt

Applications
admin, administrator, user, test

Change Passwords and Secure Settings

Step 3: Change All Default Passwords Immediately

- Never leave factory passwords in place
- Don't use simple variations of defaults (admin123 is not better than admin)
- Use strong, unique passwords for each system
- Consider using a password manager for administrative credentials
- Implement a policy that forces password changes during initial setup

Step 4: Secure Default Settings

Common settings to change:

- Default SNMP community strings (public/private)
- Default database ports (though obscurity isn't security, it adds a small layer)
- Default file sharing settings (often too permissive)
- Default firewall rules (often allow too much traffic)
- Default encryption settings (often use weak or outdated algorithms)

Maintain Configuration Consistency

Step 5: Maintain Configuration Consistency

- Apply your secure configuration standard to all new systems
- Use configuration management tools to enforce consistency
- Create golden images or templates with secure configurations baked in
- Automate deployment to avoid manual configuration errors

Step 6: Regular Configuration Audits

- Scan systems to verify they match your secure configuration standard
- Check for configuration drift (unauthorised changes)
- Review configurations after updates or changes
- Use automated tools to continuously monitor compliance

Quick Wins for Default Management

1. Change every default password on every system
2. Disable or rename default administrative accounts
3. Create a one-page quick reference guide for your secure configuration standard
4. Run monthly scans to identify systems still using defaults
5. Make secure configuration mandatory in your procurement and deployment processes

Remember: Default settings exist for convenience, not security. Your security posture is only as strong as your weakest configuration—and defaults are always the weakest.

Practice 11

Terminate Inactive Connections

What It Means: Automatically close network connections and user sessions when they're no longer actively being used. Don't leave idle sessions open indefinitely—they're opportunities for attack.

Types of Connections to Manage

User Sessions

Web applications, remote desktop, VPN connections

Database Connections

Application-to-database links

Network Connections

TCP sessions, SSH sessions, Telnet sessions

API Sessions

REST API tokens, authentication sessions

How to Implement Connection Termination

Step 1: Define Timeout Policies

Establish appropriate timeout periods based on:

- Security requirements of the system
- Sensitivity of the data being accessed
- Business needs and user convenience
- Regulatory or compliance requirements

Common Timeout Recommendations:

- High-security systems: 5-10 minutes of inactivity
- Standard business applications: 15-30 minutes
- Low-risk applications: 30-60 minutes
- Administrative interfaces: 5-10 minutes (shorter is better)
- Financial systems: 10-15 minutes
- Database connections: 5-15 minutes depending on use case

Implement Session Timeouts

Step 2: Implement Session Timeouts

For Web Applications:

- Configure idle timeout in the application settings
- Set timeout for both the session cookie and server-side session
- Display a warning before timeout (e.g., "You'll be logged out in 2 minutes")
- Require re-authentication after timeout

For Network Connections:

- Configure keep-alive timers
- Set TCP timeout values
- Enable automatic connection termination on inactivity

For Remote Access:

- Set RDP/VNC session timeout policies
- Configure VPN idle disconnect timers
- Implement SSH timeout settings

For Databases:

- Set connection pool timeout values
- Configure idle connection reaper threads
- Implement application-side connection management