D

Defense in Depth: Security Layers and Practical Examples

Defense in depth is a security strategy that combines multiple controls so one failure does not leave an entire system exposed. It includes people, operating procedures, and technology across prevention, detection, containment, and recovery. Useful layers address different failure paths and remain effective when another contr...

PUBLISHER
Gridinsoft Support

UPDATED
Oct 8, 2026

READ TIME
3 min

CONTENTS

In this guide

01 What is defense in depth?

02 Which layers belong in the design?

03 Example: a stolen email password

04 When layers share the same weakness

05 How to start and test the design

06 Defense in depth vs. zero trust

Quick answer: Defense in depth is a security strategy that combines multiple controls so one failure does not leave an entire system exposed. It includes people, operating procedures, and technology across prevention, detection, containment, and recovery. Useful layers address different failure paths and remain effective when another control is bypassed.

01 What is defense in depth?

NIST describes defense in depth as an approach integrating people, technology, and operations across multiple barriers. The practical question is: if this protection fails, what happens next? A strong design gives a specific answer for important assets, such as preventing a stolen password from becoming unrestricted access or preserving recovery copies after administrator compromise.

There is no universal requirement for exactly five or seven layers. Diagrams group controls differently. What matters is coverage of realistic attack paths and failure conditions, including dependencies between controls.

02 Which layers belong in the design?

- **People and process:** verification of sensitive requests, clear reporting, controlled changes, and incident ownership.
- **Identity:** strong authentication, limited privileges, separate administrator accounts, and session review.
- **Devices:** supported software, hardening, malware protection, and endpoint visibility.
- **Networks:** restricted access between systems, controlled remote administration, and appropriate filtering.
- **Applications and data:** secure development, authorization checks, minimization, and encryption.
- **Detection and recovery:** useful alerts, protected logs, isolated backups, and tested restoration.

03 Example: a stolen email password

Imagine a user enters a password on a phishing page. Phishing-resistant authentication may prevent the attacker from completing sign-in. If a session is nevertheless compromised, restrictions on mailbox forwarding can limit data exposure. Alerts about new sessions or rule changes can prompt investigation. Independent confirmation of a changed invoice can prevent payment fraud even if the malicious message comes from the real mailbox.

Each control has a separate job. Training reduces mistakes, authentication restricts access, monitoring reveals misuse, and financial procedures protect a business action. This example also shows why an antivirus scan alone cannot verify that a cloud account is secure.

04 When layers share the same weakness

Three tools controlled by one unrestricted account may all be disabled together. A backup on an always-connected share may be deleted with the original data. Multiple alerts that nobody reviews provide less protection than a smaller set with clear ownership. Inventory the administrator, identity provider, network path, and storage dependencies of each critical control.

Some overlap is valuable: independently detecting the same event can confirm a problem. The goal is not to eliminate overlap but to avoid paying for apparent protection that fails for exactly the same reason.

05 How to start and test the design

1. Choose a critical asset and a plausible scenario, such as account takeover or ransomware.
2. Map how an attacker could reach the asset and which controls interrupt each step.
3. Identify who maintains each control and who acts when it reports a problem.
4. Test a failure safely: use an approved account or simulation and confirm the next layer actually operates.
5. Record the evidence, remaining gap, owner, and deadline for improvement.

6. Retest after significant changes to identity, networks, suppliers, or backup systems.

06 Defense in depth vs. zero trust

Defense in depth describes combining protective measures. Zero trust emphasizes explicit access decisions and avoiding automatic trust based solely on network location. The approaches can work together: authorization at each resource can be one part of a broader layered design.

Measure outcomes such as the time to investigate a test alert, the privileges reachable with a compromised test account, and the time to restore a verified backup. A product count does not show whether the organization can contain or recover from an attack.

Reference: NIST defense-in-depth glossary. Related: multifactor authentication and data loss and recovery.

NEED MORE HELP?

We are here when the guide is not enough.

Visit the Gridinsoft Support Center for current product guidance or submit a support request with the details of your case.

Open this guide online at support.gridinsoft.com