Engineer

Information security

protecting data and systems from misuse and harm

What it is

Information security is the discipline of keeping information, and the systems that store and move it, in the state their owners intend: read only by the people meant to read it, changed only in the ways that were meant, and there when it is needed. In everyday speech, and in most job titles, it is shortened to security. The definition in NIST's glossary comes from US federal law, 44 U.S.C. 3552, which names three properties. Confidentiality preserves authorised restrictions on access and disclosure. Integrity guards against improper modification or destruction, and includes being able to show that information is authentic and that its sender cannot plausibly deny sending it. Availability means timely and reliable access to the information and use of it.

The three pull against each other, and that tension is most of what makes the work hard. A database nobody can reach is perfectly confidential and of no use to anyone. A backup kept in three places improves availability and gives a thief three places to look. Most decisions in the field are about where to sit between those pulls for one particular kind of information, and the answer for a public price list is very different from the answer for a patient's medical record.

The work is done through controls, each one a specific safeguard with a specific purpose. Some are technical: encryption, authentication, a firewall, access control. Some are procedural: a review before a change ships, a rule that departing staff lose access on their last day, a written plan for an incident. Some are physical, such as a locked room. NIST SP 800-53 is a catalogue of such controls, grouped into families and written so that an organisation can choose the ones that fit its risks. The NIST Cybersecurity Framework 2.0 works one level higher. It describes outcomes under six functions, Govern, Identify, Protect, Detect, Respond and Recover, and states that it does not prescribe how those outcomes should be achieved.

Use the concept in a decision

Put “What is information security?” to work in a real decision. We can help relate the concept to the system you are building or buying. Accounts, passwords and permissions arranged so people hold exactly the access their job needs, and leavers lose it that day.

Explore our services Get in touch →

Test the assumptions behind the term

Use the explanation and linked work to frame questions for your project. We can examine the assumptions with you before a tool is chosen. Wrote 1,000 RBAC rules to secure Linux servers — 1,000 access rules written to close a security gap

Wrote 1,000 RBAC rules to secure Linux servers Get in touch →

Why it matters

Harm to information is hard to undo. Data that has leaked stays leaked, whatever is fixed afterwards. A record that was altered without anyone noticing makes every later decision that relied on it suspect. An outage on the day a payroll run or a tender deadline falls costs more than the hours it lasts. The people whose data it was usually bear the cost first, and the company that held it second.

For a buyer, the difficult part is that security is invisible while it works. Two suppliers can list the same tools in the same words, and only one of them may have checked that those tools do what the list says. The useful questions are therefore about evidence: which controls are in place, how anyone knows they are working today rather than on the day they were installed, and what happens when one of them reports a problem. A supplier who answers with a report generated from the live system is in a different position from one who answers with a policy document.

The trade-offs are real, and they are mostly about time and friction. Every scanner produces findings, and every finding needs a person to decide whether it matters. Every extra step at login is a step people will look for ways around. Every control that is added has to be maintained, or it quietly becomes a control that exists only on paper. The Govern function opens the framework because these are management decisions as much as technical ones: someone has to decide how much risk is acceptable, which data is worth the most care, and what the rest of the protection costs.

What it looks like in practice

Security on one application platform became a build result rather than an opinion, through a continuous security programme for an application platform. Static analysis runs through CodeQL, dynamic testing through ZAP, and Go

dependencies are checked with govulncheck. A software bill of materials is generated on every build with Anchore and Syft, and Gitleaks scans history for credentials. The build's own supply chain is covered as well: the outside Actions its workflows call are fixed to exact commits, never to tags someone could move. Dependabot watches 6 ecosystems. Alongside the automation sit 21 written security audits, including a threat model and an assessment against the OWASP testing guide. A vulnerability recognised by the configured checks can fail the build. The cost is the volume of findings, and the ongoing triage it takes to keep the signal usable.

The company's own host has 464 MB of memory, too little for a metrics stack, so its security is described by seven read-only roles that render the live host into Markdown instead. The access report covers accounts, sudo scope, key fingerprints, forge users and database roles. The security report answers whether hardening is enforced rather than configured, printing the firewall rules in match order and the live ban rule set, and no number is printed that the run did not measure. Among other things, the reports found two dead security controls and an undeclared listening port.

The gap between enforced and configured had already shown itself on the same host. A rate limit sat for a year below a broader firewall rule that matched first, so no packet ever reached it. The checks now read the numbered rule listing and the live filter chain, and assert against order. Each port has exactly one rule, always with a protocol, the SSH port carries a rate limit and never an allow beside it, and the declared public surface of five ports is asserted as a whole.

Where it goes wrong

A control is configured and never enforced. A setting in a file, a rule in a list or a policy on a wiki records what someone intended. Only a check against the running system shows what is true. A programme that counts controls instead of testing them reports full coverage while the gaps stay open, and the first real test comes from someone else.

Everything rests on one layer. A strong perimeter with wide-open access behind it means a single mistake gives everything away. Defence in depth assumes that each layer will fail at some point and places another behind it, so that a stolen password still meets a second factor and a compromised service still meets a narrow database role.

Findings arrive faster than anyone reads them. A scanner that reports every possible issue at the same urgency teaches people to stop reading it. Clear owners, severity thresholds and a written rule for when a finding blocks a release keep the output worth reading.

Availability is remembered only during the outage. Confidentiality gets most of the attention because leaks make the news. Backups that have never been restored, a single server with no plan for its replacement and certificates that expire unnoticed all fail the third property, and they tend to fail it on the worst possible day.

Outsiders have no obvious place to report a problem. Someone who finds a weakness in a public service needs to know where to say so. RFC 9116 defines a security.txt file, served from /.well-known/security.txt, with a required Contact field saying where reports go and a required Expires field saying how long the file can be trusted. Without it, reports land in a general inbox or go to whoever answers first.

Further reading

- Information security, in the NIST CSRC glossary — the definition from FIPS 200 and 44 U.S.C. 3552, with its three properties.
- 44 U.S.C. 3552 — the statute that defines confidentiality, integrity and availability.
- NIST SP 800-53 Rev. 5 — the catalogue of security and privacy controls, grouped into families.
- The NIST Cybersecurity Framework 2.0 — six functions that describe security outcomes, from Govern to Recover.
- RFC 9116 — the security.txt format that tells researchers where to report a vulnerability.

Glossary

Choose for your actual constraints

Bring the deployment path, the operational risks and the people on call. We can scope a change with checks, a recovery route and clear ownership.

See what it costs Get in touch →

Make the terminology practical

Ask how “What is information security?” applies to your organisation. A concrete example from your current system is enough to start the discussion.

Explore our services

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://engineer.company/glossary/security/