



Web-Application-Security

Das Open Web Application Security Project hat dieses Jahr die OWASP Top Ten 2021 veröffentlicht, die im Vergleich zu der Liste der Sicherheitsbedrohungen im Web von 2017 einige Änderungen mitbringen. Pentests zum Aufspüren von Schwachstellen in Webanwendungen sollten gut geplant und vorbereitet werden. OAuth ersetzt die Autorisierung mit Name und Passwort durch einen tokenbasiert Ansatz und soll in Kürze in Version 2.1 erscheinen – neun Jahre nach OAuth 2.0.

ab Seite 7

Programmiersprachen

Java-Versionen erscheinen inzwischen halbjährlich, und von Java 11 bis Java 17 hat es zahlreiche Veränderungen in der Sicherheitsarchitektur gegeben. Das performante C++ ist dank freiem Speicherzugriff anfällig für Schwachstellen durch Speicherfehler, die sich mit den richtigen Maßnahmen minimieren lassen. Eine automatisierte Codeanalyse hilft beim Absichern, aber der Einsatz erfordert grundlegendes Know-how. Rust bietet interessante Konzepte, mit denen die Programmiersprache ohne Performanceeinbußen sicherer als C oder C++ ist.

ab Seite 21



Web-Application-Security

Die OWASP Top Ten 2021

Penetrationstests von Webanwendungen

OAuth 2.1 – das Sicherheitsupdate

Programmiersprachen

Sicherheitsneuerungen in Java

C++-Sicherheitsfallen und ihre Abwehr

Programmanalyse für C/C++ im Detail

Sicheres Programmieren mit Rust

Cloud-Security

Kontext als Schlüssel zur sicheren Cloud

Serverless-Security

Identity und Access Management mit Keycloak

Confidential Computing im Überblick

Container hinter Schloss und Riegel

Cloud-Compliance mit dem Open Policy Agent

Kryptografie

Kryptografie als Grundlage sicherer Software

Umgang mit kryptografischen Verfahren

Implementierung quantensicherer Kryptografie

Qualitätssicherung

8	Schutz der Schnittstellen: OWASP API Security Top 10	92
12	Ein kritischer Blick auf statische Codeanalyse-Verfahren	96
16	Automatisierte Softwaretests mit Fuzzing	102
	Privacy by Design: Vorsicht ist besser als Nachsicht	106
	Sichere Softwareentwicklung nach dem „Security by Design“-Prinzip	112
22	Versteckte Risiken beim Kompilieren von Embedded Software	116

DevSecOps

38	Grundlagen DevSecOps	122
	Marktübersicht DevSecOps-Tools	126
	Sichere Software entwickeln mit OWASP SAMM	132
44	Das Reifegradmodell Security Belts im Blick	138
48	Shift Left – Secure by Design und agile Entwicklung	144
52	Vollständiges Erfassen von Softwarelieferketten	150

Sonstiges

74		
78	Editorial	3
86	Impressum	95



Cloud-Security

Der Balanceakt zwischen Security und schnellen Entwicklungszyklen ist in der Cloud besonders schwierig. Serverless-Computing ermöglicht eine flexible Skalierung, aber Hunderte verteilte Funktionen bieten eine große Angriffsfläche, die es abzusichern gilt. Keycloak ist eine Open-Source-Anwendung für Identitäts- und Zugriffsmanagement. Confidential Computing soll mit hardwarebasierter Verschlüsselung sensibler Daten das Vertrauen in die Public Cloud stärken. Der Einsatz von Containern erfordert dedizierte Schutzmaßnahmen. Zum Durchsetzen von Compliance in der Cloud hat sich der Open Policy Agent als Allzwecktool etabliert.

ab Seite 43

Kryptografie

Zum Absichern von Software und Systemen ist Kryptografie unverzichtbar, aber das Thema ist äußerst komplex. Vor allem lauern beim praktischen Einsatz zahlreiche Stolperfallen, die sich durch gute Planung umgehen lassen. In nicht allzu ferner Zukunft werden Quantencomputer vermutlich die Karten neu mischen, da bisher als sicher geltende Algorithmen angreifbar werden.

ab Seite 73



Qualitätssicherung

Neben den OWASP Top Ten hat die Organisation eine Liste der größten Gefahren für APIs veröffentlicht. Statische Codeanalyse hilft beim Aufspüren von Fehlern und Schwachstellen, kann aber Code-Reviews nicht ersetzen. Fuzzing füttert Programme mit zufälligen oder bewusst fehlerhaften Inhalten und hat sich als Testverfahren etabliert. Neben technischen Anforderungen gilt es, beim Entwickeln die juristischen Vorgaben zum Datenschutz im Blick zu halten. Security by Design und der Security Development Lifecycle lieferten bereits vor zwölf Jahren Richtlinien für sichere Softwareentwicklung. In der Embedded-Entwicklung sind Tests auf der Zielplattform unerlässlich.

ab Seite 91



DevSecOps

Damit Security bei immer kürzeren Entwicklungszyklen nicht zu kurz kommt, bietet DevSecOps Maßnahmen, jeden Schritt von der Entwicklung bis zum Betrieb abzusichern. Inzwischen existiert eine Fülle an Tools, die bei der Umsetzung helfen. OWASP SAMM bietet ein Framework für den Secure Development Lifecycle, und die Security Belts sollen helfen, die Kompetenzen in Teams bezüglich der Sicherheit zu stärken. Shift Left steht für den Ansatz, Sicherheitsaspekte möglichst früh in den Softwarelebenszyklus einzubeziehen. Eine sichere Software Supply Chain ist im IoT-Umfeld mit Over-the-Air-Updates eine besondere Herausforderung.

ab Seite 121

