HISOLUTIONS

Know-how to go

Sicherheit in der Softwareentwicklung

Von Null auf Sicher: Ein Leitfaden zur sicheren Softwareentwicklung

André Moll

André Moll

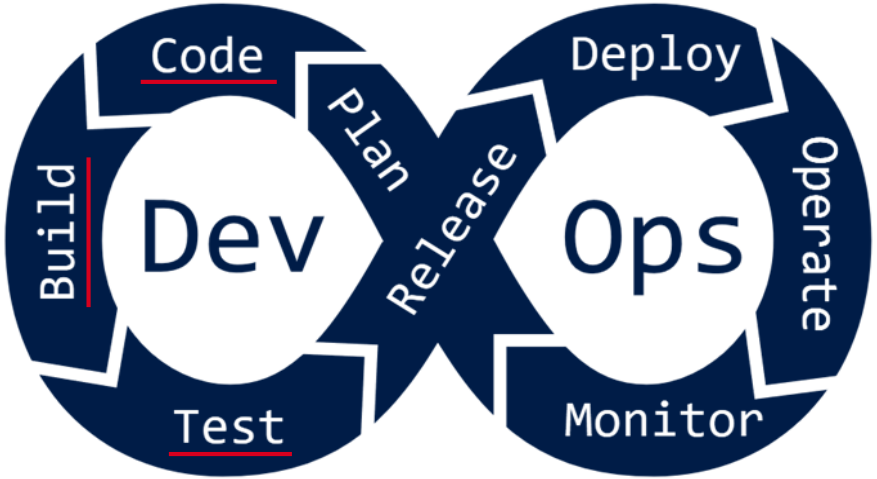
moll@hisolutions.com

- Fachliche Schwerpunkte: DevOps, Kubernetes, Azure, Cloud Transformation & Enablement
- Implementation von verschiedenen maßgenschneiderte Cloud- und DevOps-Lösungen bei unterschiedlichsten Kunden
- Ich teile gerne mein Wissen – Wissenstransfers, Vorträge, bald auch in der iX ;)

© HiSolutions 2025

Softwareentwicklung meets DevSecOps

Zugriffskontrolle



Bleiben Sie up-to-date

Clean Code

Sicheres Design

Backends und Middleware

© HiSolutions 2025

Leitfaden für sichere Softwareentwicklung

- Es gibt viele Punkte, an denen sichere Softwareentwicklung scheitern kann…
  - … sehr viele
- Nach der Anforderungsanalyse geht es ans Programmieren
- Wenn hier nicht gewissenhaft gearbeitet wird, zieht man Security Flaws in jede Stage der Software bzw. der Applikation mit
- Teilweise so tief in der Applikation, dass dies später nicht mehr erkannt werden kann
- Generell gilt: Ihre Software ist nur so sicher, wie ihr schwächstes Glied

© HiSolutions 2025

Clean Code

- Beschreiben Sie Ihre Variablen und Funktionen
- Kommentieren und dokumentieren Sie so, wie Sie es gerne vorfinden würden
- Pflegen Sie einen konsistenten Code Style
- Testen, testen, testen
- Refactoren Sie Ihren Code regelmäßig
- Arbeiten Sie mit Versionskontrolle (z. B. Git)

```
def socks5_connect(self, client_socket):
        openziti.monkeypatch()
        try:
            version, cmd, _, addr_type = struct.unpack("IMEGAHMGVK
            assert version == 5

            if cmd != 1:
                client_socket.sendall(struct.pack("!BBBBIH", 5, 7,
                raise ValueError("Only CONNECT command is supporte

            if addr_type == 1:
                address = socket.inet_ntoa(client_socket.recv(4))
            elif addr_type == 3:
                domain_length = struct.unpack("ET03", client_socke
                address = client_socket.recv(domain_length).decode
            else:
                raise ValueError("Unsupported address type")

            port = struct.unpack("!H", client_socket.recv(2))[0]

            logging.info(f"Connecting to {address}:{port}")
            remote_socket = socket.socket(socket.AF_INET, socket.S
            remote_socket.connect((address, port))

            client_socket.sendall(struct.pack("!BBBB", 5, 0, 0, 1)

            self.relay_traffic(client_socket, remote_socket)
        except Exception as e:
            logging.error(f"Error during SOCKS5 connect: {e}")
```


© HiSolutions 2025

Sicheres Design

- Nutzen Sie von Beginn an sichere Dependencies
- Benutzen Sie keine Secrets als Clear-Text in Ihrem Programmcode
  - Verwenden Sie Key-Vaults u. ä.
- Wenn Sie mit Dienstleistern arbeiten, verlangen Sie auch von diesen einen gewissen Sicherheitsstandard
- Verhindern Sie die Möglichkeit von Code-Injections
- Implementieren Sie Input Validation

© HiSolutions 2025

Beispiele für Injections (SQL)

- User Input für das Feld „Stadt“:
  Bonn
- SQL:
  SELECT * FROM OrdersTable WHERE Stadt = 'Bonn';
- User Input für das Feld „Stadt“:
  Bonn';drop table OrdersTable--
- SQL:
  SELECT * FROM OrdersTable WHERE Stadt =
  'Bonn';drop table OrdersTable--'

© HiSolutions 2025

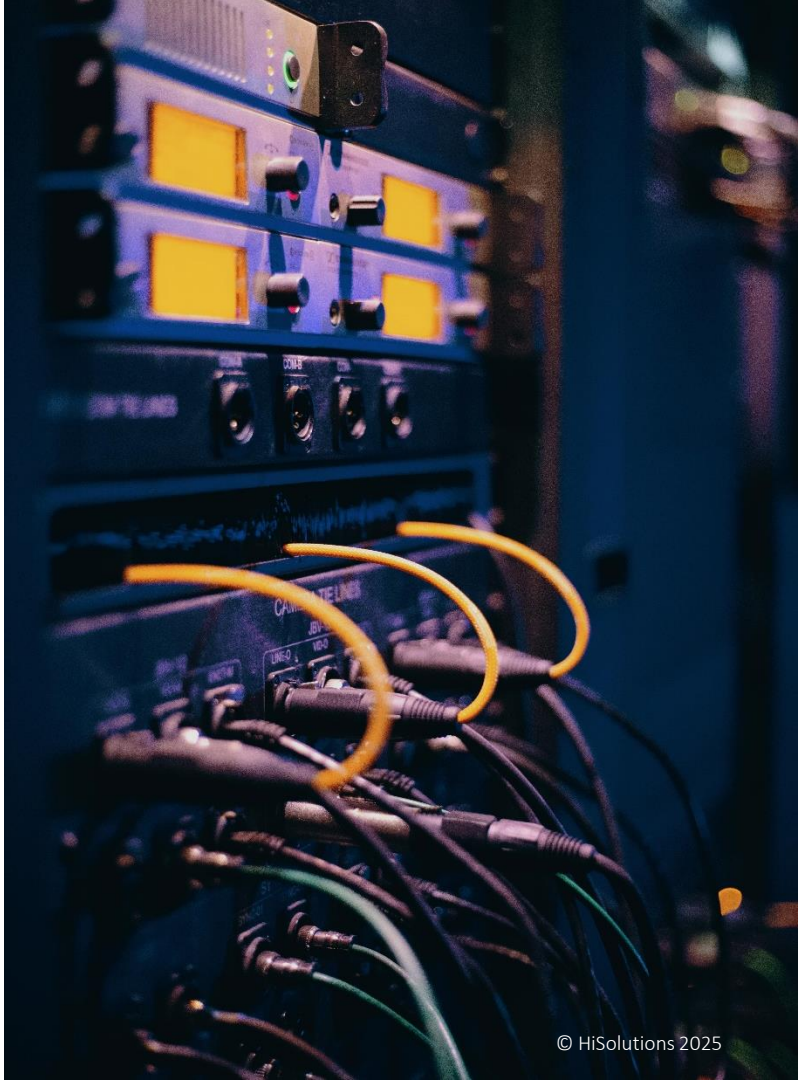
Zugriffskontrolle

- Achten Sie stets darauf, wer wann wie auf Ihre Applikation zugreifen darf
- Verwenden Sie sichere Passwörter **(auch für Dev-Umgebungen!)**
- Implementieren Sie überhaupt IAM in Ihre Software
- Überlegen Sie genau, wem Sie Zugang geben…
- … und überlegen Sie zweimal, wer was darf

© HiSolutions 2025

Backends und Middleware

- Härten Sie Ihre Systeme nach gängigen Standards
- Implementieren Sie Sicherheitsmechanismen wie u. a. RBAC (Role Based Access Control)
- Prüfen Sie, welche Webpages Sie nach außen freigeben
- Und prüfen Sie den Zweck (Beispiel SwaggerUI)
- Achten Sie auf den weiteren Betrieb und die Wartung
- Auditieren und stressen Sie regelmäßig (Drills, Pentests, Audits)



© HiSolutions 2025

Referenzen für sichere Softwareentwicklung

- OWASP (Open Worldwide Application Security Project)
  - OWASP Developer Guide (https://owasp.org/www-project-developer-guide/)
  - OWASP Top-10 (https://owasp.org/www-project-top-ten/)
- CIS (Center for Internet Security)
  - CIS-Benchmarks (https://workbench.cisecurity.org/)
    - Nicht direkt für Softwareentwicklung, aber für die Infrastruktur sehr nützlich. Behandelt unter anderem Docker, Kubernetes etc. und bietet Guides für Hardening bzw. Best-practices
- Microsoft
  - Software Development Lifecycle (https://learn.microsoft.com/de-de/compliance/assurance/assurance-microsoft-security-development-lifecycle)
- NIST (National Institute of Standards and Technology)
  - Secure Software Development Framework (https://csrc.nist.gov/projects/ssdf)

© HiSolutions 2025

HISOLUTIONS

Schloßstraße 1 | 12163 Berlin

info@hisolutions.com | +49 30 533 289 0

www.hisolutions.com

© HiSolutions GmbH