SCADIX

Jochen Weschta

Founder & CEO
SCADIX GmbH

Moderne PKI-Architekturen

Kurzlebige Zertifikate, ACME und Zero Trust in der Praxis

Wer ist SCADIX?

- **Wir** sind ein modernes IT-Consulting-Unternehmen.
- **Open Source** ist unser Werkzeug für **Innovation** und **Fortschritt**.
- **Wir** machen **Unternehmen** durch Technologie **zukunftsfähig**.
- **Unser Motor:** Leidenschaftliche IT-Expert:innen, die Innovation leben.

Important News

SCADIX

Reduktion von öffentlichen SSL/TLS Zertifikaten

15.03.2020 -> **398** days
15.03.2026 -> **200** days
15.03.2027 -> **100** days
15.03.2029 -> **47** days
10 days for domain validation

Be prepared

Rückblick in der Zeit

Zeitalter: On Premise

SCADIX

- Infrastruktur vollständig on-premise
- Service-Exposition über eine zentrale Firewall
- Verschlüsselte Kommunikation innerhalb des LANs nicht erforderlich

„Wir sind hinter der Firewall – Verschlüsselung im LAN unnötig“

Zeitalter: Neumoderne

SCADIX

- Bedarf nach Verschlüsselung auch für interne Services erkannt
- Einführung von **Microsoft Certificate Services (MSCS)** ab Windows Server 2000
- Zertifikatserstellung erfolgt oft manuell oder teilautomatisiert
- Einsatz von Zertifikaten mit **langer Laufzeit**

Zeitalter: Cloud Native

SCADIX

- Zunahme von Cloud-Diensten im Unternehmensumfeld
- Notwendigkeit zur sicheren Verbindung zwischen On-Premise und Cloud-Services
- Abschied vom Konzept des „**always trusted LANs**“

  Jede Kommunikation muss verschlüsselt erfolgen

- Verwaltung von Zertifikaten verursacht hohen Personal- und Zeitaufwand

SCADIX

Radikale Automatisierung des Zertifikatsmanagementes

Was ist ACME?

Was ist ACME?

SCADIX

- **ACME** = Automatic Certificate Management Environment
- Enwicklung: Let’s Encrypt (ACMEv1) dep. 01.07.21
- ACMEv2 offizieller Standard gemäß RFC 8555

Ziel von ACME:

- Automatisierte Ausstellung von Zertifikaten
- Verifizierung des Domainbesitzes
- Vermeidung manueller Prozesse bei der Zertifikatsbeantragung

Bedeutung Zero-Trust

Bedeutung Zero-Trust

SCADIX

- „**Vertraue niemandem – jede Anfrage muss verifiziert werden.**“
- ACME validiert jede Zertifikatsanfrage automatisch (z. B. HTTP-01, DNS-01).
- **Menschliches Vertrauen** wird durch **technische Verifikation ersetzt** – Kernprinzip von Zero-Trust.

Extern vs Intern

Extern vs Intern

SCADIX

Externe Zertifikate:

- Verwendet für **öffentliche** Webservices, APIs, Mail-Server etc.
- Von **öffentlich** anerkannten Certificate Authorities (CAs) signiert
- Sicherstellung der Vertrauenswürdigkeit im **Internet**

Interne Zertifikate:

- Für die Kommunikation **innerhalb** des Unternehmensnetzwerks
- Häufig von einer **internen** Certificate Authorities (CA) ausgestellt
- Ermöglicht verschlüsselte Kommunikation **ohne** öffentliches Vertrauen

Zertifikat Livecycle intern

Zertifikat Livecycle intern

SCADIX

Manuelle Prozesse:

- Zertifikate werden oft über Microsoft Certificate Services (MSCS) erstellt, die **nicht ACME-fähig** sind
- Prozess oft **manuell** oder **teilautomatisiert** mit Shell-Skripten
- **Human Gates verhindern** einen **vollständig** automatisierten Ablauf

Zertifikat Livecycle intern

SCADIX

Benötigte Rollen:

- **Person 1:** Admin der CA (Zertifikate ausstellen)
- **Person 2:** Applikationsexperte und Ersteller des Signing Requests (CSR)
- **Person 3:** Admin, der das Zertifikat ausrollt, in die Anwendung integriert und den Applikationsservice neu startet

Was ist ein CSR?

CSR = Certificate Signing Request

- **Zentraler Bestandteil** der **Zertifikatserstellung** zur Beantragung eines signierten Zertifikats bei einer **CA**

- CSR-Files können von **verschiedenen** Nutzern erstellt werden und enthalten alle nötigen **Eigenschaften** für die **Zertifikatsanforderung**

Eigenschaft	Beschreibung
Öffentlicher Schlüssel	Entspricht dem privaten Schlüssel
Common Name (CN)	Domain oder Servername (z. B. example.com)
Organization (O)	Name der Organisation
Organizational Unit (OU)	Abteilung innerhalb der Organisation
Country (C)	Land des Antragstellers
State (ST)	Bundesland oder Region
Locality (L)	Stadt oder Ort
Subject Alternative Names (SANs)	Optionale zusätzliche Namen oder IP-Adressen
Signatur	Mit dem privaten Schlüssel signierte Anfrage

Zertifikat Livecycle intern

SCADIX

Herausforderungen:

- Notwendigkeit für **mindestens 2-3** Fachkräfte (Applikationsexperte, Automatisierungsadmin, Zertifikatsmanager)
- Manuelle Schritte (**Human Gates**) verlängern den Prozess erheblich
- **Sicherheitsaspekte** werden nicht ausreichend geprüft
- Wer überprüft, ob das Zertifikat korrekt verwendet wird und nicht missbräuchlich auf anderen Servern?

SCADIX

ineffizient
und
riskant

Zertifikat Livecycle extern

Zertifikat Livecycle extern

SCADIX

Fortgeschrittene automatisierte Workflows:

- Der Einsatz von **ACME Client's** wie "**certbot**" nimmt Einzug, durch den Einsatz von freien **Let's Encrypt** Zertifikaten.

- **Let's Encrypt:**
  - Kostenlos, automatisiert und Open Source
  - Eingeschränkte Funktionen und Support

- **SECTIGO/Globalsign:**
  - Umfassender Service, inklusive erweiterten Funktionen, Support und Compliance
  - Bevorzugt durch Unternehmen mit spezifischen Geschäftsanforderungen und Compliance-Vorgaben

SCADIX

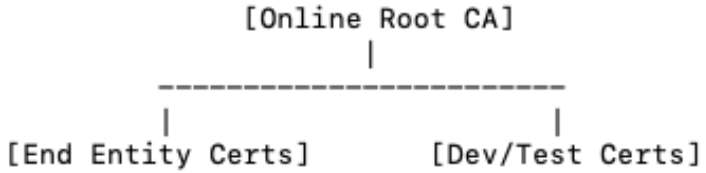
Wie erhalte ich ähnlich elegante Lösung in meiner internen Infrastruktur?

Best Practice PKI Infrastruktur

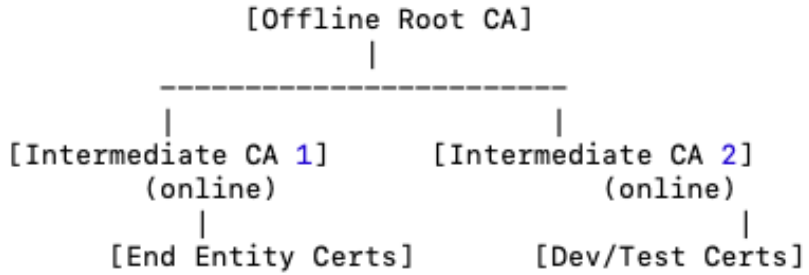
Best Practice PKI

SCADIX

Nicht empfohlen

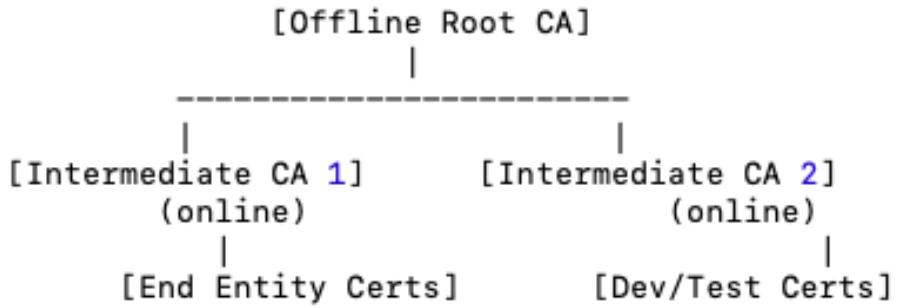


Empfohlen



Unterschied online/offline CA?

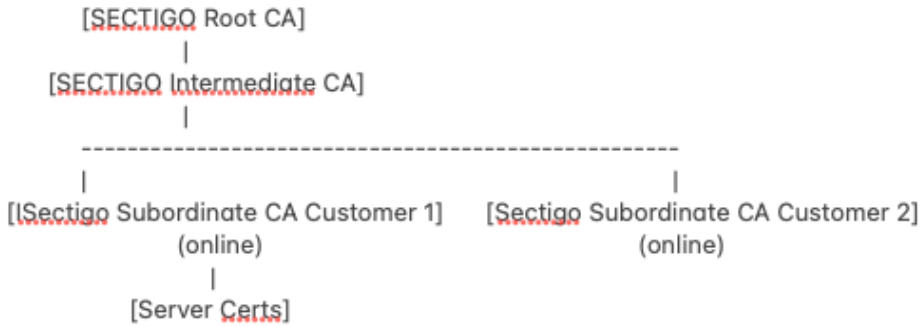
Kategorie	Online-CA	Offline-CA
Erreichbarkeit	Netzwerkbasiert, automatisiert	nur manuelle Nutzung
Sicherheit	Mittel	Hoch
Flexibilität	Hoch (API, ACME etc.)	Gering
Einsatzzweck	Zertifikatsausstellung, Renewal	Root-Vertrauen, Signatur von CAs



Hosted (Sub)ordinate CA?

Vorteile:

- Verantwortung für Root CA entfällt.
- Zertifikate gelten “out of the box” als sicher, da über die OS-Packages die Root CA’s verteilt werden.



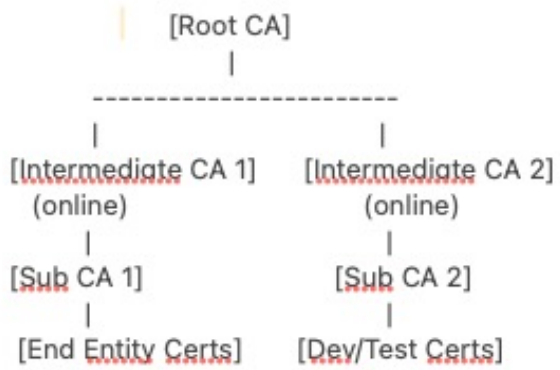
Was ist denn der Unterschied?

Sub CA vs Intermediate CA?

Unterschied: Sub-CA vs Intermediate

SCADIX

Technisch ist es das gleiche.



Begriff	Bedeutung
Intermediate CA	Eine **Zertifizierungsstelle**, deren Zertifikat **von einer Root CA** signiert wurde
Subordinate CA (Sub-CA)	Einfach ein **untergeordneter CA-Knoten** – also **jede CA unterhalb der Root**, egal ob Intermediate oder tiefer

SCADIX

Unabhängig und Eigenverantwortlich

Was ist die STEPCA?

Was ist die STEPCA?

- OpenSource Lösung **smallstep.**
- Ein leichtgewichtiges **CA-System** für das Management von **Zertifikaten**
- Vollständig in **Go** geschrieben, verfügbar als **Einzelbinary** oder **Container.**

SCADIX

Funktionen
X.509 Certificate Authority
SSH Certificate Authority
Provisioners
Templates
Cryptographic Protection

Integration in die Infrastruktur

SCADIX

- **StepCA** wird in einem **Teilbereich** der Infrastruktur (Single- oder HA-Setup) platziert.
- Stellt eine **REST-API** bereit, über die **Clients** mit der CA **kommunizieren**.
- Das Admin-Tool "**step**" ermöglicht die **Interaktion** mit der Online-CA.
- Unterstützt den **ACMEv2**-Standard und kann als **Alternative** zu "**certbot**" verwendet werden.
- Eignet sich für die **automatische** Ausstellung und **Verwaltung** von Zertifikaten in **internen** Umgebungen.

Neuaufbau oder Migration

Neuaufbau

SCADIX

```
#> step certificate create --profile root-ca "SCADIX Root CA"
root_ca.crt root_ca.key
```

```
#> step certificate create scadix.org scadix.crt scadix.key \\
    --profile leaf --not-after=8760h \\
    --ca ./intermediate_ca.crt --ca-key ./intermediate_ca.key --bundle
```

By using the --bundle flag we automatically bundle the new leaf certificate with the signing intermediate certificate. TLS-based services will require the bundle in order to verify the full chain.

Migration

SCADIX

#> **$ step ca init --root=[ROOT_CERT_FILE] --key=[ROOT_PRIVATE_KEY_FILE]**

Vorteil:

- Parallelbetrieb zu alternativen CA’s (z.B. MSCS) möglich
- Kein harter Switch notwendig.

Provisioner

Provisioner in StepCA

SCADIX

- **Zweck:** Authentifizierung & Autorisierung von Zertifikatsanforderungen
- **Typen:**
  - JWK – JSON Web Key-basierte Authentifizierung
  - OAuth/OIDC – Single Sign-on via Identity Provider
  - X5C – Basierend auf vorhandenen X.509-Zertifikaten
  - SSHPOP – SSH Certificate Proof-of-Possession
  - **ACME – Automatisierte Zertifikatserneuerung (z.B. via Certbot)**
  - Nebula – Integration mit Nebula Overlay-Netzwerken
  - SCEP – Unterstützung für Simple Certificate Enrollment Protocol
  - K8sSA – Authentifizierung über Kubernetes Service Accounts
  - Cloud Provisioners – z.B. AWS, GCP, Azure Instanz-Metadaten

ACME Challenges

SCADIX

Merkmal	HTTP-01	DNS-01
Nachweis über	Datei auf `http://<domain>/.well-known/acme-challenge/<token>`	DNS-TXT-Eintrag `_acme-challenge.<domain> = <token>`
Erfordert	Öffentlichen HTTP-Zugriff (Webserver)	DNS-Zugriff bzw. API
Typische Nutzung	Webseiten-Zertifikate	Wildcard-Zertifikate, interne Hosts
Wildcard Support	❌ Kein Wildcard-Support	✅ Wildcard-Zertifikate möglich
Verifizierung	IP/Hostname	Hostname über DNS
Vorteil	Einfacher für öffentlich erreichbare Webserver	Kein Webserver notwendig, flexibler für interne/externe Systeme

Es gibt noch weitere Challenges als HTTP-01 und DNS-01

Herausforderungen HTTP-01 Challenge

SCADIX

- Port **80** wird bereits vom Webserver für den redirect auf Port **443** verwendet
- Ausnahme kann definieren werden, sodass der **ACME** Token über den bestehenden Webserver exposed wird.

```
server {
    listen 80;
    server_name example.com;

    location /.well-known/acme-challenge/ {
        root /var/www/html;
    }

    location / {
        return 301 https://$host$request_uri;
    }
}
```

Praxis

ACME basierte Zertifikats-erstellung

SCADIX

```
#> step ca certificate --provisioner acme scadix.org scadix.crt scadix.key

✓ Provisioner: acme (ACME)
Using Standalone Mode HTTP challenge to validate scadix.org. done!
Waiting for Order to be 'ready' for finalization .. done!
Finalizing Order .. done!
✓ Certificate: scadix.crt
✓ Private Key: scadix.key
```

Zertifikat Insights

SCADIX

```
#> step certificate inspect --short foo.crt
X.509v3 TLS Certificate (ECDSA P-256) [Serial: 1664...3445]
  Subject:     scadix.org
  Issuer:      Speedy Intermediate CA
  Provisioner: admin [ID: w1OU...oZUg]
  Valid from:  2024-05-01T21:15:16Z
          to:  2024-05-02T21:15:16Z
```

Renewal der Zertifikate


```
#> $ step ca renew --force scadix.crt scadix.key
Your certificate has been saved in foo.crt

$ step certificate inspect --short foo.crt
X.509v3 TLS Certificate (ECDSA P-256) [Serial: 1664...3445]
  Subject:     scadix.org
  Issuer:      Speedy Intermediate CA
  Provisioner: admin [ID: w1OU...oZUg]
  Valid from:  2024-05-01T21:15:16Z
          to:  2024-05-02T21:15:16Z
```

Autorenewal der Zertifikate

SCADIX

- Renewal using systemd

…

ExecCondition=/usr/bin/step certificate needs-renewal ${CERT_LOCATION}

; ExecStart renews the certificate, if ExecStartPre was successful.

ExecStart=/usr/bin/step ca renew --force ${CERT_LOCATION} ${KEY_LOCATION}

; Try to reload or restart the systemd service that relies on this cert-renewer

ExecStartPost=/usr/bin/env sh -c "! systemctl --quiet is-active %i.service || systemctl try-reload-or-restart %i"

- …

SCADIX

Abbau des ganzen manuellen fehleranfälligen Workflows.

Policy/Sicherheitsmaßnahmen

SCADIX

- Zertifikate bleiben bis zum **Ablaufdatum gültig**, sofern die Intermediate CA nicht widerrufen wird.
- **Worst Case:** Kompromittiertes Wildcard-Zertifikat mit langer Laufzeit → erfordert Widerruf der Intermediate CA → alle zugehörigen Zertifikate werden ungültig.

Empfehlung:

- **Keine** Wildcard-Zertifikate mit **langer** Laufzeit verwenden.
- Einsatz **mehrerer Intermediates** mit gemeinsamem Root.
- **Besser:** Short-Term-Zertifikate mit vollständiger Automatisierung – **sicherer, einfacher, eleganter.**

Reduktion der Laufzeit

SCADIX

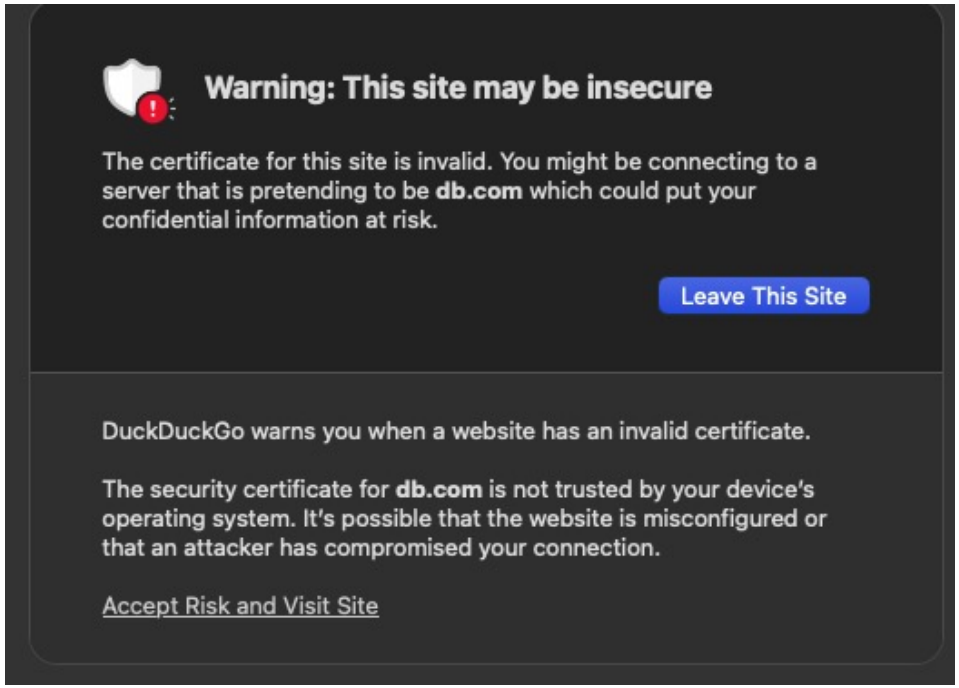
- Reduktion der Laufzeit kann zentral über den Provisionier konfiguriert werden.

```
#> step ca provisioner update acme-provisioner \
 --default-tls-cert-duration=72h \
 --max-tls-cert-duration=168h
```

Vertrauensfrage

Problem:

SCADIX



Vertrauensherstellung

SCADIX

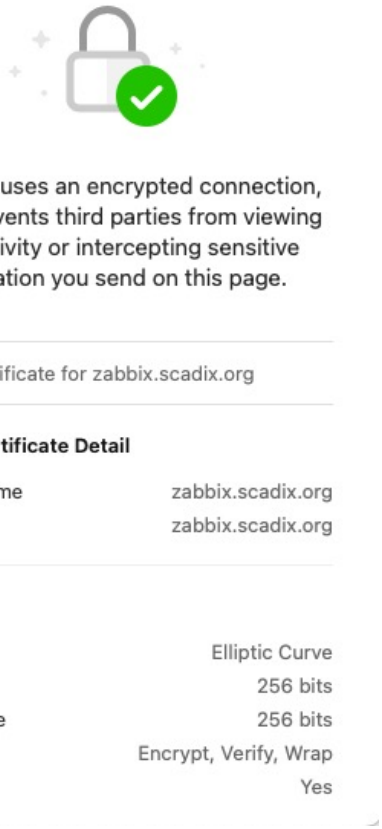
- **Aufbau Initiale** Vertrauensherstellung zwischen einem **Client** und der **CA** durch Abruf und Speicherung des CA-Zertifikats

=> Bootstrapping

(nur notwendig, wenn man eine komplett eigene self signed Root CA verwendet. Bei der Verwendung des Hosted SubCA ist dies nicht erforderlich.

```
#> step ca bootstrap --ca-url https://ca.scadix.org:9000 --fingerprint
8e91a6ad6ac4983eec193f12f0d5525b40dabaa0b7d026ebba03d17efd010b97 --install
```

Finally:


This page uses an encrypted connection, which prevents third parties from viewing your activity or intercepting sensitive information you send on this page.
Certificate for zabbix.scadix.org
Security Certificate Detail
Common Name zabbix.scadix.org
Summary zabbix.scadix.org
Public Key
Algorithm Elliptic Curve
Key Size 256 bits
Effective Size 256 bits
Usage Encrypt, Verify, Wrap
Permanent Yes

SCADIX

Wer setzt noch MSCS (Microsoft Certificate Service) ein?

Connect

SCADIX

Persönlich auf Veranstaltungen

https://scadix.de

jochen.weschta@scadix.de

[m] jochen.weschta:matrix.scadix.de

https://github.com/scadix-gmbh

https://www.linkedin.com/company/scadix

SCADIX

Jochen Weschta

Founder & CEO
SCADIX GmbH

Vielen Dank!

und weiterhin viel Spaß auf dem 21. Linux Info Tag!