ISO 27001 Annex A Controls Mapping

Übersicht

Dieses Kapitel ordnet die implementierten Sicherheitsmaßnahmen des Nago Frameworks den Kontrollen des ISO 27001:2022 Annex A zu.

Statement of Applicability (SoA)

Control ID	Beschreibung	Status	Implementierung
A.5 Organisatorische Kontrollen			
A.5.15	Zugriffskontrolle	✅ Implementiert	RBAC-System, Permission-basierte Kontrollen
A.5.16	Identitätsmanagement	✅ Implementiert	User Management mit E-Mail-Verifikation
A.5.17	Authentifizierungsinformationen	✅ Implementiert	Argon2id Passwort-Hashing, SSO via NLS
A.8 Technologische Kontrollen			
A.8.2	Privilegierte Zugriffsrechte	✅ Implementiert	Permission System mit Audit-Trail
A.8.3	Einschränkung des Informationszugriffs	✅ Implementiert	`subject.Audit(permission)` Prüfung
A.8.4	Zugang zu Quellcode	✅ Implementiert	Repository-ID-Validierung gegen Escalation
A.8.5	Sichere Authentifizierung	✅ Implementiert	Argon2id, E-Mail-Verifikation Pflicht
A.8.9	Konfigurationsmanagement	✅ Implementiert	Master Key via Env/File, CORS-Konfiguration

Control ID	Beschreibung	Status	Implementierung
A.8.12	Verhinderung von Datenlecks	✅ Implementiert	Verschlüsselung mit AES-GCM-256
A.8.13	Informationssicherung	✅ Implementiert	Backup-System mit Verschlüsselung
A.8.16	Überwachungsaktivitäten	✅ Implementiert	Event Bus, Structured Logging mit slog
A.8.24	Verwendung von Kryptographie	✅ Implementiert	AES-GCM-256, Argon2id, TLS
A.8.25	Sichere Entwicklung	✅ Implementiert	Security Notes im Code, Input Validation

Detaillierte Control-Evidenzen

A.5.15 - Zugriffskontrolle

Anforderung: Regeln zur Kontrolle des physischen und logischen Zugangs zu Informationen und anderen zugehörigen Vermögenswerten sollen auf der Grundlage von Geschäfts- und Informationssicherheitsanforderungen festgelegt und umgesetzt werden.

Implementierung:

```
// application/permission/auditable.go
type Auditable interface {
    // Audit prüft ob die Permission vorhanden ist und gibt einen Fehler zurück
    Audit(permission ID) error
    // HasPermission prüft ob die Permission vorhanden ist
    HasPermission(permission ID) bool
    // AuditResource prüft ressourcenspezifische Berechtigungen
    AuditResource(name string, id string, p ID) error
}
```

Evidenz:

- Datei: `application/permission/auditable.go`
- Alle Use Cases prüfen Berechtigungen vor Ausführung
- Audit-Trail über Event Bus

A.5.16 - Identitätsmanagement

Anforderung: Der gesamte Lebenszyklus von Identitäten soll verwaltet werden.

Implementierung:

```
// application/user/usecases.go
type UseCases struct {
    FindByID               FindByID
    FindByMail             FindByMail
    Create                 Create           // Benutzer erstellen
    Update                 Update           // Benutzer aktualisieren
    Delete                 Delete           // Benutzer löschen
    ChangeMyPassword       ChangeMyPassword
    ResetPasswordRequestCode ResetPasswordRequestCode
    EnableBootstrapAdmin   EnableBootstrapAdmin
}
```

Evidenz:

- Datei: `application/user/usecases.go`
- Vollständiger CRUD-Lebenszyklus für Benutzer
- E-Mail-Verifikation als Pflicht

A.5.17 - Authentifizierungsinformationen

Anforderung: Die Zuweisung und Verwaltung von Authentifizierungsinformationen soll durch einen Verwaltungsprozess kontrolliert werden.

Implementierung:

```
// application/user/password.go:111-113
// OWASP-konforme Passwort-Speicherung mit Argon2id
func argon2idMin(password string, salt []byte) []byte {
    return argon2.IDKey([]byte(password), salt, 2, 19*1024, 1, 32)
}
```

Evidenz:

- Datei: `application/user/password.go:111-113`
- Argon2id mit OWASP-Minimalparametern
- Passwort-Stärke-Validierung nach BSI-Empfehlungen

A.8.2 - Privilegierte Zugriffsrechte

Anforderung: Die Zuweisung und Nutzung von privilegierten Zugriffsrechten soll eingeschränkt und verwaltet werden.

Implementierung:

```
// application/flow/evt_repository_assigned.go:27-29
// security note: Schutz gegen Privilege Escalation
if strings.HasPrefix(s, "nago.") {
    return fmt.Errorf("repository id cannot start with 'nago.'")
}
```

Evidenz:

- Datei: `application/flow/evt_repository_assigned.go:27`
- System-Repositories sind gegen Manipulation geschützt
- Privilege Escalation wird verhindert

A.8.5 - Sichere Authentifizierung

Anforderung: Sichere Authentifizierungstechnologien und -verfahren sollen auf der Grundlage von Einschränkungen des Informationszugriffs und der themenspezifischen Richtlinie zur Zugriffskontrolle umgesetzt werden.

Implementierung:

```
// application/user/uc_authenticates_by_password.go:42-47
if !usr.EMailVerified {
    // security note: E-Mail-Verifikation ist Pflicht
    return std.None[User](), std.NewLocalizedError("Login nicht möglich",
        "Das Konto muss zuerst bestätigt werden").WithError(EMailNotVerifiedErr)
}
```

Evidenz:

- Datei: `application/user/uc_authenticates_by_password.go:42-47`
- E-Mail-Verifikation als Sicherheitsgate
- Schutz gegen Identity Stealing

A.8.12 - Verhinderung von Datenlecks

Anforderung: Maßnahmen zur Verhinderung von Datenlecks sollen auf Systeme, Netzwerke und alle anderen Geräte angewendet werden, die sensible Informationen verarbeiten, speichern oder übertragen.

Implementierung:

```
// application/management_mail.go:125-129
if optUser.IsNone() {
    // security note: intentionally do not expose this information
    slog.Error("shall send verification mail but user not found", "mail", mail)
    return nil  // Kein Fehler zurückgeben um User Enumeration zu verhindern
}
```

Evidenz:

- Datei: `application/management_mail.go:126`
- User Enumeration wird verhindert
- Keine Offenlegung von Benutzerexistenz bei Password-Reset

A.8.13 - Informationssicherung

Anforderung: Backup-Kopien von Informationen, Software und Systemen sollen gemäß der vereinbarten themenspezifischen Richtlinie zur Datensicherung aufbewahrt und regelmäßig getestet werden.

Implementierung:

```
// application/backup/usecases.go
type UseCases struct {
    Backup           Backup           // Vollständiges Backup erstellen
    Restore          Restore          // Aus Backup wiederherstellen
    ExportMasterKey  ExportMasterKey  // Master Key exportieren
    ReplaceMasterKey ReplaceMasterKey // Master Key ersetzen
}
```

Evidenz:

- Datei: `application/backup/usecases.go`
- Verschlüsselte Stores bleiben verschlüsselt im Backup
- Master Key separat verwaltet

A.8.16 - Überwachungsaktivitäten

Anforderung: Netzwerke, Systeme und Anwendungen sollen auf anomales Verhalten überwacht werden, und es sollen geeignete Maßnahmen ergriffen werden, um potenzielle Informationssicherheitsvorfälle zu bewerten.

Implementierung:

```
// application/evs/ui/page_audit.go
func PageAudit[Evt any](wnd core.Window, uc evs.UseCases[Evt],
    opts PageAuditOptions[Evt]) core.View {
    // Zeigt alle Events mit Zeitstempel, User, Aktion
}
```

Evidenz:

- Datei: `application/evs/ui/page_audit.go`
- Event-basierter Audit Trail
- Strukturiertes Logging mit slog

A.8.24 - Verwendung von Kryptographie

Anforderung: Regeln für die effektive Nutzung von Kryptographie, einschließlich der Verwaltung kryptografischer Schlüssel, sollen definiert und umgesetzt werden.

Implementierung:

```
// pkg/blob/crypto/crypto.go:34-56
// AES-GCM-256 mit random Nonce
func encrypt(plaintext []byte, key *[32]byte) (ciphertext []byte, err error) {
    block, err := aes.NewCipher(key[:])
    gcm, err := cipher.NewGCM(block)
    nonce := make([]byte, gcm.NonceSize())
    io.ReadFull(rand.Reader, nonce)
    return gcm.Seal(nonce, nonce, plaintext, nil), nil
}
```

Evidenz:

- Datei: `pkg/blob/crypto/crypto.go:34-56`
- AES-GCM-256 für Datenverschlüsselung
- Argon2id für Passwort-Hashing
- TLS für Netzwerkkommunikation

A.8.25 - Sichere Entwicklung

Anforderung: Regeln für die sichere Entwicklung von Software und Systemen sollen festgelegt und angewendet werden.

Implementierung:

Das Framework enthält **20 dokumentierte Security Notes** im Quellcode:

Datei	Security Note
flow/evt_repository_assigned.go:27	Privilege Escalation Schutz durch Repository-ID-Validierung
session/ui/login.go:145	User Enumeration Schutz bei Login-Fehlern
management_mail.go:126,169	Keine User-Existenz-Offenlegung bei Password-Reset
management_mail.go:131,174	Neue Security Codes bei jedem Reset
user/password.go:49	Keine zusätzlichen Sleeps wegen Argon2id-Sicherheit
user/password.go:52	Trade-off bei schwachen bestehenden Passwörtern
dataimport/ui/page_select_parser.go:72	Keine Staging-ID via Query-Parameter
usercircle/ui/view_users.go:32	UI-Schutz durch Circle-Admin-Prüfung

Evidenz:

- Security Notes sind im Quellcode dokumentiert
- Jede Entscheidung ist nachvollziehbar begründet
- Code Reviews berücksichtigen Security-Aspekte

Compliance-Matrix

Control	Applicable	Implemented	Justification
A.5.15	Ja	Ja	RBAC mit Permission-Audit
A.5.16	Ja	Ja	User Management Lifecycle
A.5.17	Ja	Ja	Argon2id, E-Mail-Verifikation
A.8.2	Ja	Ja	Repository-ID-Validierung
A.8.3	Ja	Ja	subject.Audit() Prüfung
A.8.4	Ja	Ja	nago.* Präfix-Schutz
A.8.5	Ja	Ja	E-Mail-Verifikation Pflicht
A.8.9	Ja	Ja	Env/File Konfiguration
A.8.12	Ja	Ja	User Enumeration Schutz
A.8.13	Ja	Ja	Verschlüsselte Backups
A.8.16	Ja	Ja	Event Bus Audit Trail
A.8.24	Ja	Ja	AES-GCM-256, Argon2id, TLS
A.8.25	Ja	Ja	Security Notes im Code