STRIDE Bedrohungsmodell

Übersicht

Diese Analyse verwendet das STRIDE-Modell zur systematischen Identifikation von Bedrohungen für das Nago Framework. Jede Bedrohungskategorie wird mit implementierten Mitigationen dokumentiert.

STRIDE-Kategorien

```
@startuml
title STRIDE Bedrohungsmodell - Nago Framework

rectangle "Bedrohungen" {
    rectangle "Spoofing" as S #ffcccc
    rectangle "Tampering" as T #ffddcc
    rectangle "Repudiation" as R #ffffcc
    rectangle "Information\nDisclosure" as I #ccffcc
    rectangle "Denial of\nService" as D #ccccff
    rectangle "Elevation of\nPrivilege" as E #ffccff
}

rectangle "Komponenten" {
    rectangle "Session\nManagement" as SM
    rectangle "User\nManagement" as UM
    rectangle "Permission\nSystem" as PM
    rectangle "Blob\nStorage" as BS
    rectangle "Backup\nSystem" as BU
    rectangle "WebSocket" as WS
}

S --> SM : Session Hijacking
S --> UM : Identity Theft
T --> BS : Data Manipulation
T --> BU : Backup Tampering
R --> PM : Audit Evasion
I --> BS : Data Leakage
I --> BU : Master Key Exposure
D --> UM : Brute Force
D --> WS : Connection Flooding
E --> PM : Permission Bypass
E --> UM : Privilege Escalation

@enduml
```

Spoofing (Identitätsverschleierung)

S-01: Session Hijacking

Bedrohung	**Angreifer stiehlt Session-Cookie und übernimmt Benutzersitzung**
Wahrscheinlichkeit	Mittel
Auswirkung	Hoch - Vollständiger Zugriff auf Benutzerkonto
Mitigationen	• HttpOnly Cookie verhindert JavaScript-Zugriff • Secure Flag erzwingt HTTPS-Übertragung • SameSite=Strict verhindert CSRF • Session-TTL von 3 Monaten begrenzt Exposure-Fenster
Status	✅ Mitigiert

S-02: Token Theft via API

Bedrohung	**Angreifer extrahiert API-Token und authentifiziert sich**
Wahrscheinlichkeit	Niedrig
Auswirkung	Hoch - API-Zugriff mit Token-Berechtigungen
Mitigationen	• API-Tokens werden mit Argon2id gehasht gespeichert • Reverse Hash Lookup ohne Salt (Performance-Trade-off) • Token-Rotation möglich
Code-Referenz	`application/user/password.go:97-108`
Status	✅ Mitigiert

S-03: User Enumeration bei Login

Bedrohung	**Angreifer erkennt gültige Benutzerkonten durch unterschiedliche Fehlermeldungen**
Wahrscheinlichkeit	Mittel
Auswirkung	Niedrig - Information für Brute-Force-Vorbereitung
Mitigationen	• Einheitliche Fehlermeldung: "Der Benutzer existiert nicht, das Konto wurde deaktiviert oder das Kennwort ist falsch." • Security Note in `session/ui/login.go:145`
Code-Referenz	`application/session/ui/login.go:145`
Status	✅ Mitigiert

Tampering (Datenmanipulation)

T-01: MITM-Angriff auf Kommunikation

Bedrohung	**Angreifer manipuliert Daten während der Übertragung**
Wahrscheinlichkeit	Niedrig (mit TLS)
Auswirkung	Hoch - Datenintegrität kompromittiert
Mitigationen	• TLS für alle HTTP-Verbindungen (Deployment-Ebene) • TLS für SMTP mit Zertifikatsvalidierung • WebSocket über WSS
Code-Referenz	`application/mail/smtp.go:28-31`
Status	✅ Mitigiert (Deployment-abhängig)

T-02: Backup-Manipulation

Bedrohung	**Angreifer manipuliert Backup-Datei und führt kompromittierte Daten ein**
Wahrscheinlichkeit	Niedrig
Auswirkung	Kritisch - Systemübernahme möglich
Mitigationen	• Verschlüsselte Stores bleiben verschlüsselt • Master Key separat gespeichert • Backup-Restore nur mit Berechtigung
Code-Referenz	`application/backup/usecases.go`
Status	⚠️ Teilweise mitigiert - Keine kryptographische Signatur

T-03: Repository-Daten-Manipulation

Bedrohung	**Angreifer mit Flow-Rechten manipuliert System-Repositories**
Wahrscheinlichkeit	Mittel
Auswirkung	Kritisch - Privilege Escalation
Mitigationen	• Repository-IDs dürfen nicht mit `nago.` beginnen • Validierung bei Zuweisung
Code-Referenz	`application/flow/evt_repository_assigned.go:27`
Status	✅ Mitigiert

Repudiation (Abstreitbarkeit)

R-01: Audit-Log-Manipulation

Bedrohung	**Angreifer löscht oder manipuliert Audit-Logs**
Wahrscheinlichkeit	Niedrig
Auswirkung	Hoch - Forensik nicht möglich
Mitigationen	• Event Store als Append-Only-Struktur • Events enthalten Zeitstempel und User-ID • Zugriff auf Event Store nur mit Berechtigung
Code-Referenz	`application/evs/`
Status	Mitigiert

R-02: Aktions-Abstreitbarkeit

Bedrohung	**Benutzer bestreitet durchgeführte Aktion**
Wahrscheinlichkeit	Mittel
Auswirkung	Mittel - Compliance-Probleme
Mitigationen	• Alle sicherheitsrelevanten Aktionen werden geloggt • Event-Envelope enthält User-ID, Zeitstempel, Aktion • Strukturiertes Logging mit slog
Code-Referenz	`application/evs/ui/page_audit.go`
Status	Mitigiert

Information Disclosure (Informationsoffenlegung)

I-01: Master Key Exposure

Bedrohung	**Angreifer erhält Zugriff auf Master Key**
Wahrscheinlichkeit	Niedrig
Auswirkung	Kritisch - Alle verschlüsselten Daten kompromittiert
Mitigationen	• Umgebungsvariable `NAGO_MASTER_KEY` (empfohlen) • Datei `.masterkey` mit 0600 Permissions (Fallback) • Master Key nicht im Backup enthalten

Bedrohung	**Angreifer erhält Zugriff auf Master Key**
Code-Referenz	`application/security.go:27-67`
Status	✅ Mitigiert

I-02: User Enumeration bei Password-Reset

Bedrohung	**Angreifer erkennt gültige E-Mail-Adressen**
Wahrscheinlichkeit	Mittel
Auswirkung	Niedrig - Information für Phishing
Mitigationen	• Keine Unterscheidung ob User existiert oder nicht • Gleiche Antwort bei existierendem und nicht-existierendem User
Code-Referenz	`application/management_mail.go:126`
Status	✅ Mitigiert

I-03: User Existenz bei Registrierung

Bedrohung	**Angreifer erkennt registrierte E-Mail-Adressen**
Wahrscheinlichkeit	Hoch
Auswirkung	Niedrig - Akzeptierter Trade-off
Mitigationen	• Bewusste Design-Entscheidung dokumentiert • Keine Sleep-Verzögerung wegen Mutex-Lock • Alternative: Captcha (nicht implementiert)
Code-Referenz	`application/user/uc_create.go:130`
Status	⚠️ Akzeptiertes Risiko

I-04: Backup-Datei-Leak

Bedrohung	**Backup-Datei wird unbefugt zugänglich**
Wahrscheinlichkeit	Niedrig
Auswirkung	Hoch - Sensible Daten exponiert
Mitigationen	• Verschlüsselte Stores bleiben verschlüsselt • Master Key separat gespeichert • Unverschlüsselte Daten: Risikoanalyse erforderlich
Code-Referenz	`application/backup/usecases.go`
Status	⚠️ Teilweise mitigiert

Denial of Service (Dienstverweigerung)

D-01: Brute-Force-Login

Bedrohung	**Angreifer versucht Passwörter durch Massenanfragen zu erraten**
Wahrscheinlichkeit	Hoch
Auswirkung	Mittel - Kontoübernahme bei schwachem Passwort
Mitigationen	• Argon2id macht Brute-Force teuer (Memory-Hard) • Passwort-Stärke-Validierung erzwingt starke Passwörter • ⚠️ Kein globales Rate-Limiting implementiert
Code-Referenz	`application/user/password.go:111`
Status	⚠️ Teilweise mitigiert - Rate-Limiting fehlt

D-02: User-Erstellung-DoS

Bedrohung	**Angreifer überflutet System mit Registrierungsanfragen**
Wahrscheinlichkeit	Mittel
Auswirkung	Mittel - System wird langsam
Mitigationen	• Mutex-Lock erlaubt nur 1 User/Sekunde • E-Mail-Verifikation verhindert Spam-Accounts
Code-Referenz	`application/user/uc_create.go:31-32`
Status	✅ Mitigiert

D-03: Password-Hashing-DoS

Bedrohung	**Angreifer sendet extrem lange Passwörter**
Wahrscheinlichkeit	Mittel
Auswirkung	Mittel - CPU/Memory-Erschöpfung
Mitigationen	• Maximale Passwortlänge: 1000 Zeichen • Validierung vor Hashing
Code-Referenz	`application/user/password.go:167`
Status	✅ Mitigiert

D-04: WebSocket Connection Flooding

Bedrohung	**Angreifer öffnet viele WebSocket-Verbindungen**
Wahrscheinlichkeit	Mittel
Auswirkung	Hoch - Server-Ressourcen erschöpft
Mitigationen	• ⚠️ Keine Verbindungslimits implementiert • Deployment-Ebene: Reverse Proxy Limits
Status	⚠️ Nicht mitigiert auf Application-Ebene

Elevation of Privilege (Rechteausweitung)

E-01: Permission Bypass

Bedrohung	**Angreifer umgeht Berechtigungsprüfungen**
Wahrscheinlichkeit	Niedrig
Auswirkung	Kritisch - Unbefugter Zugriff
Mitigationen	• Alle Use Cases verwenden `subject.Audit(permission)` • Fehlende Berechtigung gibt Error zurück • Audit-Trail für alle Zugriffe
Code-Referenz	`application/permission/auditable.go`
Status	✅ Mitigiert

E-02: Repository-ID-Escalation

Bedrohung	**Angreifer mit Flow-Rechten übernimmt System-Repositories**
Wahrscheinlichkeit	Mittel
Auswirkung	Kritisch - System-Kompromittierung
Mitigationen	• Repository-IDs dürfen nicht mit `nago.` beginnen • Validierung bei jeder Repository-Zuweisung • Security Note dokumentiert
Code-Referenz	`application/flow/evt_repository_assigned.go:27`
Status	✅ Mitigiert

E-03: Bootstrap-Admin-Missbrauch

Bedrohung	**Bootstrap-Admin wird für persistenten Zugriff missbraucht**
Wahrscheinlichkeit	Niedrig
Auswirkung	Kritisch - Administratorzugriff

Bedrohung	Bootstrap-Admin wird für persistenten Zugriff missbraucht
Mitigationen	• Zeitlimit für Bootstrap-Admin • Automatische Deaktivierung nach Ablauf • Audit-Log für alle Admin-Aktionen
Code-Referenz	`application/management_user.go`
Status	Mitigiert

E-04: User Circle Bypass

Bedrohung	Benutzer greift auf Ressourcen außerhalb seines Circles zu
Wahrscheinlichkeit	Mittel
Auswirkung	Mittel - Unbefugter Ressourcenzugriff
Mitigationen	• UI prüft Circle-Admin-Status bei jedem Render • Sofortiger Exit bei fehlendem Zugriff • Security Note dokumentiert
Code-Referenz	`application/usercircle/ui/view_users.go:32`
Status	Mitigiert

Risiko-Matrix

```
@startuml
title Risiko-Matrix

scale 600 width

rectangle "Kritisch" as C #ff0000
rectangle "Hoch" as H #ff8800
rectangle "Mittel" as M #ffff00
rectangle "Niedrig" as L #00ff00

note right of C
  E-01, E-02, E-03
  I-01, T-02
end note

note right of H
  S-01, S-02
  D-04, I-04
end note

note right of M
  S-03, D-01
```

```
  E-04, R-02
end note

note right of L
  T-01, R-01
  D-02, D-03
end note

@enduml
```

Zusammenfassung der offenen Punkte

ID	Bedrohung	Risiko	Empfehlung
D-01	Brute-Force-Login	Mittel	Globales Rate-Limiting implementieren
D-04	WebSocket Flooding	Hoch	Connection Limits auf Application-Ebene
T-02	Backup-Manipulation	Mittel	Kryptographische Signatur für Backups
I-03	User Enumeration bei Registrierung	Niedrig	Captcha oder alternative UX erwägen
WebSocket	Origin-Validierung	Hoch	CheckOrigin-Funktion implementieren (TODO in Code)