AYI NEDJIMI Consultants

Bypass FIDO2 et Passkeys : Attaques sur l'Authentification

Catégorie : Articles Techniques Lecture : 7 min Publié le : 15/02/2026 Auteur : Ayi NEDJIMI

Contournement de FIDO2/WebAuthn : AitM sur l'enrollment, device binding bypass, token theft. Thèmes : passkeys, authentication bypass, MFA bypass.

Cette analyse detaillee de Bypass FIDO2 et Passkeys : Attaques sur l'Authentification s'appuie sur les retours d'experience d'equipes de securite confrontees quotidiennement aux menaces actuelles. Les methodologies presentees couvrent l'ensemble du cycle de vie de la securite, de la detection initiale a la remediation complete, en passant par l'investigation forensique et le durcissement des configurations. Les recommandations sont directement applicables dans les environnements de production et tiennent compte des contraintes operationnelles rencontrees par les equipes techniques sur le terrain. Les outils et techniques presentes ont ete valides dans des contextes reels d'incidents et de tests d'intrusion. La mise en oeuvre d'une strategie de defense en profondeur reste essentielle face a l'evolution constante du paysage des menaces, en combinant prevention, detection et capacite de reponse rapide aux incidents de securite.

Cette analyse technique de Bypass FIDO2 et Passkeys : Attaques sur l'Authentification s'appuie sur les retours d'experience d'equipes confrontees quotidiennement aux defis operationnels du domaine. Les methodologies presentees couvrent l'ensemble du cycle de vie, de la conception initiale au deploiement en production, en passant par les phases de test et de validation. Les recommandations sont directement applicables dans les environnements professionnels.

Table des matières

Auteur : Ayi NEDJIMI **Date :** 15 février 2026

Notre avis d'expert

Le Security by Design est souvent invoqué, rarement pratiqué. Intégrer la sécurité dès la conception coûte 6 fois moins cher que de corriger en production. Nos audits d'architecture montrent que les choix techniques des premières sprints conditionnent la posture de sécurité pour des années.

Introduction FIDO2/WebAuthn

FIDO2 (Fast IDentity Online) et son protocole web WebAuthn représentent la promesse d'une authentification sans mot de passe, résistante au phishing et aux attaques de replay. Adoptés par Apple (Passkeys), Google et Microsoft, ces standards sont présentés comme la solution définitive aux problèmes d'authentification. En 2026, plus de 2 milliards d'appareils supportent les passkeys, et les principaux fournisseurs d'identité (Entra ID, Okta, Duo) les proposent comme méthode d'authentification principale.

Cependant, aucun système de sécurité n'est invulnérable. Si FIDO2/WebAuthn élimine effectivement le phishing de credentials traditionnel, de nouveaux vecteurs d'attaque émergent : compromission de la phase d'enrollment, contournement du device binding, vol de tokens de session post-authentification, et exploitation des mécanismes de recovery. Cet article analyse ces vecteurs d'attaque en profondeur, avec des démonstrations techniques et des recommandations de durcissement pour les architectes sécurité et les red teamers.

Combien de vos contrôles de sécurité ont été testés en conditions réelles cette année ?

Architecture et promesses de sécurité

Le protocole WebAuthn

WebAuthn repose sur la cryptographie asymétrique. Lors de l'enrollment, l'authenticator (clé de sécurité, TPM, Secure Enclave) génère une paire de clés publique/privée. La clé publique est envoyée au Relying Party (serveur), tandis que la clé privée reste protégée dans l'authenticator et ne quitte jamais l'appareil. Lors de l'authentification, le serveur envoie un challenge que l'authenticator signe avec la clé privée.

```
// Flux WebAuthn Registration (simplifié)
// 1. Le serveur génère les options
const publicKeyCredentialCreationOptions = {
    challenge: crypto.getRandomValues(new Uint8Array(32)),
    rp: {
        name: "Acme Corp",
        id: "acme.com"  // Relying Party ID = domaine
    },
    user: {
        id: Uint8Array.from("user123", c => c.charCodeAt(0)),
        name: "alice@acme.com",
        displayName: "Alice"
    },
    pubKeyCredParams: [
        { alg: -7, type: "public-key" },   // ES256
        { alg: -257, type: "public-key" }  // RS256
    ],
    authenticatorSelection: {
        authenticatorAttachment: "platform",  // ou "cross-platform"
        residentKey: "required",  // Passkey (discoverable credential)
        userVerification: "required"  // Biométrie/PIN obligatoire
    },
    timeout: 60000,
    attestation: "direct"  // Attestation de l'authenticator
};

// 2. Le navigateur appelle l'authenticator
const credential = await navigator.credentials.create({
    publicKey: publicKeyCredentialCreationOptions
});

// 3. L'authenticator :
//    a) Vérifie que rpId correspond à l'origine (anti-phishing)
//    b) Génère une paire de clés (privée stockée dans le TPM/SE)
//    c) Signe l'attestation avec la clé privée
//    d) Retourne : publicKey + credentialId + attestation

// Protection anti-phishing :
// L'authenticator lie la credential au rpId (domaine)
// Un site de phishing sur evil.com ne peut pas obtenir
// une signature valide pour acme.com
```

Passkeys vs Security Keys

Caractéristique	Security Keys (FIDO2)	Passkeys (Synced)
Stockage clé privée	Hardware dédié (YubiKey, Titan)	Keychain cloud (iCloud, Google Password Manager)
Synchronisation	Non (single device)	Oui (multi-device via cloud)
Résistance au vol device	Très forte (PIN + hardware)	Dépend de la sécurité du compte cloud
Résistance AitM	Forte (origin binding)	Forte (origin binding)
Recovery	Clé de backup physique	Recovery du compte cloud
Surface d'attaque	Faible (hardware isolé)	Plus large (cloud sync, multi-device)

Cas concret

L'attaque sur SolarWinds Orion (2020) a illustré les limites des architectures de sécurité traditionnelles. L'insertion d'une backdoor dans le processus de build du logiciel a contourné toutes les couches de défense, rappelant que la supply-chain logicielle est un vecteur de menace de premier ordre.

Adversary-in-the-Middle sur l'Enrollment

Attaque sur la phase d'enregistrement

FIDO2 protège l'authentification, mais la phase d'enrollment (enregistrement initial de la passkey) reste vulnérable si l'utilisateur n'est pas encore protégé par FIDO2. L'attaque consiste à intercepter la session pendant l'enrollment via un proxy AitM (Adversary-in-the-Middle) comme Evilginx2 ou Modlishka, puis à enregistrer l'authenticator de l'attaquant à la place de celui de la victime.

```
# Scénario d'attaque AitM sur l'enrollment FIDO2

# Phase 1 : Setup du proxy AitM (Evilginx2)
# Le proxy intercepte la session d'enrollment
phishlets hostname login.acme.com login-acme.phishing.com
phishlets enable acme-enrollment

# Phase 2 : Phishing de l'enrollment
# Email : "Votre entreprise migre vers l'authentification
# sans mot de passe. Enregistrez votre passkey ici :"
# Lien : https://login-acme.phishing.com/enroll

# Phase 3 : L'utilisateur clique et s'authentifie
# via le proxy (mot de passe + MFA legacy capturés)
# Le proxy obtient une session authentifiée

# Phase 4 : Au lieu de passer le WebAuthn challenge
# au navigateur de la victime, le proxy :
# a) Bloque la réponse WebAuthn de la victime
# b) Soumet son propre authenticator au serveur
# c) L'attaquant enregistre SA passkey sur le compte victime

# Phase 5 : L'attaquant peut maintenant s'authentifier
# avec sa propre passkey sur le vrai site acme.com
# La victime ne peut plus s'authentifier (sa passkey
# n'a jamais été enregistrée)
```

Pourquoi cette attaque fonctionne
WebAuthn vérifie l'origin (rpId) mais pas l'identité de la personne effectuant l'enrollment. Si l'attaquant a accès à une session authentifiée (via AitM sur le mot de passe), il peut enregistrer n'importe quel authenticator. La protection anti-phishing de FIDO2 ne s'applique qu'après l'enrollment -- pas pendant.

Push Bombing combiné au FIDO2 enrollment

Une variante combine le push bombing (envoi massif de notifications MFA) avec l'enrollment FIDO2. L'attaquant compromet les credentials (via leak, spray, etc.), déclenche des demandes de MFA en boucle, et quand l'utilisateur finit par accepter (fatigue MFA), l'attaquant initie immédiatement l'enrollment d'une passkey malveillante. Cette technique a été observée dans des campagnes APT ciblant des entreprises tech en 2025.

Device Binding Bypass

Compromission du Keychain cloud (Synced Passkeys)

Les passkeys synchronisées (Apple iCloud Keychain, Google Password Manager, Windows Hello) introduisent un vecteur d'attaque inexistant avec les security keys hardware : la compromission du compte cloud qui héberge les clés privées. Si un attaquant compromet le compte iCloud/Google d'un utilisateur, il obtient accès à toutes ses passkeys synchronisées.

```
# Scénario : compromission iCloud pour vol de passkeys

# 1. L'attaquant compromet le compte iCloud de la victime
# Via phishing AitM, SIM swap, ou social engineering du support Apple

# 2. L'attaquant active iCloud Keychain sur son propre appareil
# Les passkeys sont synchronisées automatiquement

# 3. L'attaquant peut maintenant utiliser les passkeys de la victime
# Sur son propre appareil, pour accéder à tous les comptes
# protégés par passkeys

# Impact : TOUTES les passkeys synchronisées sont compromises
# En une seule attaque, l'attaquant obtient accès à :
# - Comptes bancaires
# - Email professionnel
# - Réseaux sociaux
# - Services cloud
# Tout ce qui utilisait des passkeys iCloud

# Détection : surveiller les événements d'enrollment
# de nouveaux appareils dans iCloud
# Alerter sur la synchronisation de Keychain vers un
# nouvel appareil non reconnu
```

Extraction de clés depuis le TPM/Secure Enclave

Bien que les clés FIDO2 soient protégées par des éléments matériels (TPM, Secure Enclave), des attaques side-channel avancées ont démontré la possibilité d'extraire des clés cryptographiques. Les attaques par analyse de puissance (DPA/SPA), les attaques par faute (fault injection) et les attaques par timing peuvent théoriquement compromettre des TPM. En pratique, ces attaques nécessitent un accès physique prolongé et un équipement spécialisé, mais elles sont réalistes pour des acteurs étatiques.

```
# Attaques connues sur les TPM et authenticators

# CVE-2023-21937 : fTPM (firmware TPM) extraction
# Les TPM logiciels (fTPM dans AMD/Intel) sont moins
# résistants que les TPM discrets
# Attaque : cold boot + extraction de la clé depuis la RAM
# avant que le fTPM la nettoie

# ROCA (CVE-2017-15361) : clés RSA faibles dans Infineon TPM
# Les clés RSA générées par certains TPM Infineon
# étaient factorisables en quelques heures
# Impact : YubiKey 4 (avant firmware 4.3.5)

# Attaque par glitching sur YubiKey 5 NFC
# Démontré par NinjaLab (2024) :
# Injection de fautes électromagnétiques pendant
# l'opération ECDSA pour récupérer la clé privée
# Nécessite : accès physique, oscilloscope, ~1 heure
# Impact : clone complet de la security key

# Mitigation : utiliser des authenticators certifiés FIDO L2+
# qui résistent aux attaques side-channel de base
```

Votre processus de patch management couvre-t-il l'ensemble de votre parc applicatif ?

Token Theft Post-Authentification

Le maillon faible : les tokens de session

La plus grande limitation de FIDO2 est qu'il ne protège que la phase d'authentification. Une fois l'utilisateur authentifié, la session est maintenue par des cookies ou tokens JWT classiques, qui sont tout aussi vulnérables au vol qu'avant FIDO2. Un attaquant n'a pas besoin de contourner FIDO2 s'il peut voler le cookie de session après l'authentification.

```
# Attaque AitM post-FIDO2 avec Evilginx2

# Configuration Evilginx2 pour capture de token post-FIDO2
# Le proxy laisse passer l'authentification FIDO2 complète
# puis capture le cookie de session résultant

# 1. La victime visite le site de phishing
# 2. Le proxy redirige vers le vrai site (transparent)
# 3. La victime s'authentifie avec sa passkey (succès !)
# 4. Le serveur émet un cookie de session
# 5. Le proxy intercepte le cookie AVANT qu'il n'atteigne
#    le navigateur de la victime
# 6. L'attaquant importe le cookie dans son navigateur

# evilginx2 phishlet pour capture post-FIDO2
auth_tokens:
  - domain: '.acme.com'
    keys: ['session_id', '__Host-session', 'auth_token']
    # Capturer ces cookies après l'authentification FIDO2

# Résultat : l'attaquant a une session authentifiée valide
# FIDO2 a été correctement complété, mais la session
# qui en résulte est volée

# Impact : accès complet au compte de la victime
# jusqu'à expiration du token (souvent 24h-30j)
```

Vol de token via malware (token theft)

```
# Extraction de cookies de session depuis Chrome (post-FIDO2)
# Même après authentification FIDO2, les cookies sont
# stockés dans le profil Chrome et extractibles

# Windows : les cookies Chrome sont dans SQLite
# chiffrés avec DPAPI (déchiffrable par le même user)
$cookieDb = "$env:LOCALAPPDATA\Google\Chrome\User Data\Default\Cookies"

# Linux : chiffrement AES avec clé dérivée du Keyring
# Outil : cookie-extractor, SharpChrome, etc.

# Attaque plus élaborée : Browser-in-the-Browser (BitB)
# Créer une fausse fenêtre de navigateur pour capturer
# le résultat de l'authentification FIDO2
# L'utilisateur pense utiliser son authenticator sur le vrai site
# mais le token de session est intercepté par l'iframe malveillant

# Mitigation critique : Token Binding / DPoP
# Lier le token de session au certificat TLS ou à une clé
# cryptographique du navigateur
# Rend les tokens volés inutilisables sur un autre appareil
```

Attaques sur le Recovery

Le paradoxe du recovery FIDO2

Le recovery (récupération de compte après perte de l'authenticator) est le talon d'Achille des déploiements FIDO2. Si un utilisateur perd sa security key ou son téléphone, il doit pouvoir récupérer l'accès à son compte. Mais les mécanismes de recovery réintroduisent exactement les vecteurs d'attaque que FIDO2 était censé éliminer :

- **Recovery par email** : Vulnérable au phishing, compromission de boîte mail
- **Recovery par SMS** : Vulnérable au SIM swap, interception SS7
- **Recovery par support humain** : Vulnérable au social engineering
- **Recovery codes** : Vulnérables au vol physique, screenshot, malware
- **Recovery par un collègue/admin** : Vulnérable à l'insider threat, compromission admin

```
# Attaque sur le recovery flow d'Entra ID avec FIDO2

# 1. L'attaquant identifie que la cible utilise FIDO2-only
# via l'enumération des méthodes d'auth disponibles

# 2. L'attaquant appelle le support IT de l'entreprise
# "J'ai perdu ma YubiKey et je suis bloqué hors de mon compte"

# 3. Le helpdesk vérifie l'identité via :
# - Questions de sécurité (facilement recherchées)
# - Vérification du badge employé (photo forgée)
# - Approbation du manager (compromis par pretexting)

# 4. Le helpdesk lance le Temporary Access Pass (TAP) :
# POST https://graph.microsoft.com/v1.0/users/{id}/\
#   authentication/temporaryAccessPassMethods
# {
#   "startDateTime": "2026-02-15T00:00:00Z",
#   "lifetimeInMinutes": 60,
#   "isUsableOnce": false
# }

# 5. L'attaquant utilise le TAP pour s'authentifier
# et enregistrer sa propre security key FIDO2

# 6. Accès complet au compte, contournement total de FIDO2

# Détection : alerter sur les événements de TAP creation
# et les enrollments FIDO2 suivant un reset de MFA
```

Durcissement et bonnes pratiques

Sécuriser l'enrollment

Recommandations pour l'enrollment FIDO2

- Effectuer l'enrollment initial en personne (physiquement) ou via un canal vérifié (vidéo avec pièce d'identité)
- Exiger une MFA forte pré-existante avant de permettre l'enrollment FIDO2 (éviter le bootstrap problem)
- Imposer un délai de 24-48h entre la demande d'enrollment et son activation (cooling period)
- Notifier l'utilisateur ET le manager de tout nouvel enrollment de security key
- Limiter le nombre de security keys par utilisateur (2-3 max) et journaliser chaque enrollment
- Utiliser l'attestation pour vérifier le modèle de l'authenticator (blocage des authenticators non approuvés)

Protéger les tokens post-authentification

```
# Implémenter Token Binding / DPoP (RFC 9449)
# Lie le token de session à une clé cryptographique du client

# DPoP (Demonstrating Proof of Possession) :
# 1. Le client génère une paire de clés éphémère
# 2. À chaque requête, le client signe un proof JWT
# 3. Le serveur vérifie que le token est utilisé par
#    le même client qui l'a obtenu

# Exemple DPoP header :
# DPoP: eyJ0eXAiOiJkcG9wK2p3dCIsImFsZyI6IkVTMjU2Iiwi...
# {
#   "typ": "dpop+jwt",
#   "alg": "ES256",
#   "jwk": { ... }  // Clé publique éphémère
# }
# Payload :
# {
#   "jti": "unique-id",
#   "htm": "POST",
#   "htu": "https://acme.com/api/resource",
#   "iat": 1708992000,
#   "ath": "hash-du-access-token"
# }

# Configuration Entra ID : Continuous Access Evaluation (CAE)
# Révoque les tokens en temps réel lors d'événements critiques :
# - Changement de mot de passe
# - Désactivation du compte
# - Changement de localisation réseau
# - Nouveau device enrollment

# Configuration Entra ID : Token Protection Policy
# Lie les tokens à un certificat d'appareil (Windows Hello)
# Rend les tokens volés inutilisables sur d'autres appareils
```

Sécuriser le recovery

- **Deux security keys minimum** : Exiger l'enregistrement d'au moins 2 security keys (une principale, une de backup stockée en lieu sûr)
- **Recovery supervisé** : Le reset de MFA doit nécessiter l'approbation de 2 personnes minimum (manager + sécurité)
- **Temporary Access Pass contrôlé** : Durée minimale (1h), usage unique, journalisation complète, notification immédiate
- **Pas de fallback vers MFA faible** : Ne jamais permettre le fallback vers SMS/email si FIDO2 est la méthode principale
- **Vérification d'identité renforcée** : Appel vidéo avec pièce d'identité et vérification du badge pour les resets

---

Questions frequentes

Comment ce sujet impacte-t-il la securite des organisations ?

Ce sujet a un impact significatif sur la securite des organisations car il touche aux fondamentaux de la protection des systemes d'information. Les entreprises doivent evaluer leur exposition, mettre en place des mesures preventives adaptees et former leurs equipes pour faire face aux risques associes a cette problematique.

Quelles sont les bonnes pratiques recommandees par les experts ?

Les experts recommandent une approche basee sur les risques, incluant l'evaluation reguliere de la posture de securite, la mise en place de controles techniques et organisationnels, la formation continue des equipes et l'adoption des referentiels de securite reconnus comme ceux du NIST, de l'ANSSI et de l'OWASP.

Pourquoi est-il important de se former sur ce sujet en 2026 ?

En 2026, la maitrise de ce sujet est devenue incontournable face a l'evolution constante des menaces et des exigences reglementaires. Les professionnels de la cybersecurite doivent maintenir leurs competences a jour pour proteger efficacement les actifs numeriques de leur organisation et repondre aux obligations de conformite.

Pour approfondir ce sujet, consultez notre outil open-source security-automation-framework qui facilite l'automatisation des workflows de sécurité.

Conclusion

FIDO2 et les passkeys représentent une avancée majeure dans la sécurité de l'authentification. Ils éliminent effectivement le phishing de credentials traditionnel et les attaques de replay de mots de passe. Cependant, ils ne constituent pas une solution miracle : les attaquants s'adaptent en ciblant les phases non protégées du cycle de vie de l'authentification -- l'enrollment, le recovery, et la session post-authentification.

Les principales conclusions de cette analyse sont :

- FIDO2 déplace la surface d'attaque mais ne l'élimine pas. Les attaquants ciblent désormais l'enrollment, le recovery et les tokens de session.
- Les passkeys synchronisées (iCloud, Google) offrent une meilleure expérience utilisateur mais introduisent le risque de compromission du compte cloud.
- Le vol de token post-authentification (via AitM ou malware) contourne complètement FIDO2. Token Binding et DPoP sont essentiels.
- Le recovery est le talon d'Achille de tout déploiement FIDO2. Des procédures de recovery robustes sont aussi importantes que le FIDO2 lui-même.
- La combinaison security key hardware + Token Binding + CAE + recovery supervisé offre le niveau de sécurité le plus élevé actuellement disponible.

Sources et références : MITRE ATT&CK · CERT-FR

Ressources et références

- Abus OAuth/OIDC : Consent Grant, Device Code, Token Replay
- Phishing Sans Pièce Jointe
- Azure AD Applications Enregistrées
- Évasion EDR/XDR

Ayi NEDJIMI

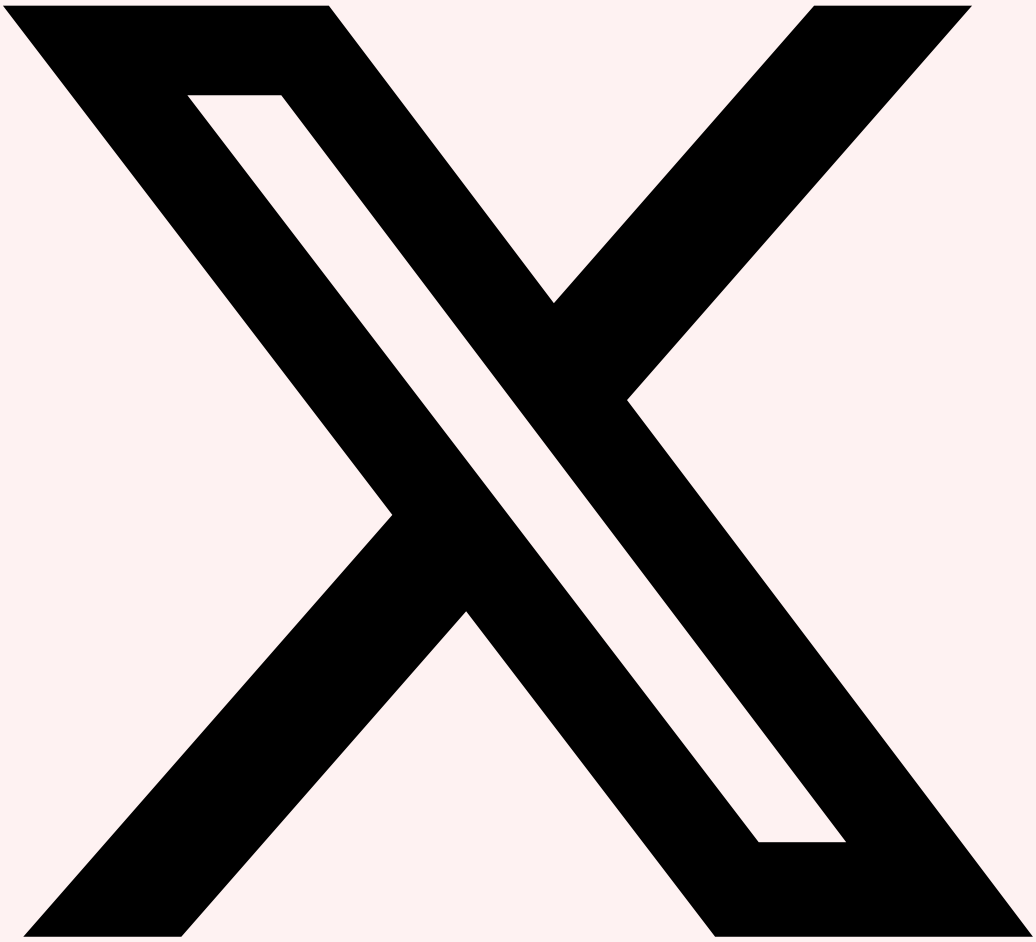
Expert en Cybersécurité & Intelligence Artificielle

Consultant senior avec plus de 15 ans d'expérience en sécurité offensive, audit d'infrastructure et développement de solutions IA. Certifié OSCP, CISSP, ISO 27001 Lead Auditor et ISO 42001 Lead Implementer. Intervient sur des missions de pentest Active Directory, sécurité Cloud et conformité réglementaire pour des grands comptes et ETI.

LinkedIn Profil complet Tous ses articles

Partagez cet Article

Cet article vous a été utile ? Partagez-le avec votre réseau professionnel !



Partager sur X

Partager sur LinkedIn

Références et ressources externes

- OWASP Testing Guide — Guide de référence pour les tests de sécurité web
- MITRE ATT&CK T1556.006 — Multi-Factor Authentication Interception
- PortSwigger Academy — Ressources d'apprentissage en sécurité web
- CWE — Common Weakness Enumeration — catalogue de faiblesses logicielles
- NVD — National Vulnerability Database — base de vulnérabilités du NIST

Ayi NEDJIMI Consultants — Expert cybersécurité offensive & intelligence artificielle

ayinedjimi-consultants.fr · ayi@ayinedjimi-consultants.fr