

Dix défis majeurs dans la modélisation ontologique de cybersécurité : Une analyse simplifiée selon la taxonomie de Bloom

Franck Jeannot

Montréal, Canada, 14 Novembre 2025, AB828E, v1.1

Abstract

Le développement d'ontologies formelles pour la cybersécurité représente un défi important à l'ère de l'ingénierie des connaissances assistée par l'IA. Cet article présente une analyse de dix défis fondamentaux rencontrés dans la conception, l'implémentation et la maintenance d'ontologies de cybersécurité. On examine ces défis à travers le prisme de la Taxonomie de Bloom, les catégorisant par complexité cognitive du niveau application au niveau création. L'analyse synthétise les connaissances issues des principaux référentiels de cybersécurité incluant l'Unified Cyber Ontology (UCO), Common Vulnerabilities and Exposures (CVE), Common Weakness Enumeration (CWE), Common Attack Pattern Enumeration and Classification (CAPEC), et MITRE ATT&CK. Une analyse technique détaillée couvrant l'évolution temporelle, l'ambiguïté sémantique, l'interopérabilité inter-standards, la gestion de l'incertitude, la complexité computationnelle et l'alignement humain-IA est faite. Bien que les systèmes d'IA puissent générer des ontologies syntaxiquement valides, des défis fondamentaux en cohérence sémantique, raisonnement contextuel et maintenance dynamique demeurent des problèmes de recherche ouverts. Cette analyse fournit des perspectives actionnables pour les chercheurs et praticiens développant les systèmes de représentation des connaissances en cybersécurité de prochaine génération.

Keywords: Ontologies de cybersécurité, Modélisation de vulnérabilités, Ingénierie des connaissances, Langage Naturel Contrôlé, OWL, Développement ontologique assisté par IA, MITRE ATT&CK, CVE, CWE, Taxonomie de Bloom

1. Introduction

La croissance exponentielle des cybermenaces et la complexité croissante des systèmes d'information ont créé un besoin urgent de représentation structurée et interprétable par la machine des connaissances en cybersécurité. Les ontologies sont apparues comme une approche prometteuse pour formaliser les connaissances en cybersécurité, permettant le raisonnement automatisé, l'intégration de threat intelligence, et l'interopérabilité entre outils et frameworks de sécurité hétérogènes.

Les avancées récentes en Modèles de Langage de Grande Taille (LLM) et en IA générative ont introduit de nouvelles possibilités pour l'ingénierie ontologique automatisée. Cependant, l'application d'approches assistées par IA au développement d'ontologies de cybersécurité révèle des défis fondamentaux qui couvrent les dimensions techniques, sémantiques et épistémologiques. Ces défis sont particulièrement aigus dans les ontologies centrées sur les vulnérabilités, qui doivent capturer des paysages de menaces en évolution rapide tout en maintenant la cohérence logique et la tractabilité computationnelle.

Cet article présente une analyse systématique de dix défis critiques dans l'ingénierie ontologique de cybersécurité, organisés selon la Taxonomie de Bloom de la complexité cognitive. L'analyse est fondée sur le développement et l'évaluation d'une ontologie complète de cybersécurité. Cette ontologie théorique intègre de multiples standards internationaux incluant CVE/NVD, CWE, CAPEC, MITRE ATT&CK, OWASP Top 10, et ISO/IEC 27001 :2022.

1.1. Contexte et Motivation

Le domaine de la cybersécurité se caractérise par sa nature hautement dynamique : plus de 20 nouvelles vulnérabilités CVE sont publiées quotidiennement, MITRE ATT&CK ajoute des centaines de techniques chaque année, et de nouveaux vecteurs d'attaque émergent constamment. Cette volatilité pose des défis uniques pour les approches traditionnelles de modélisation ontologique, qui présument généralement des domaines relativement stables.

De plus, la cybersécurité implique de multiples parties prenantes avec des besoins d'information variés : analystes SOC nécessitant des détections en temps réel, chercheurs en sécurité explorant de nouvelles classes de vulnérabilités, auditeurs évaluant la conformité réglementaire, et développeurs intégrant des pratiques de sécurité. Une ontologie efficace doit servir ces audiences diverses tout en maintenant la cohérence et la précision.

1.2. Contributions

Les contributions principales de cet article sont :

1. Une taxonomie de dix défis majeurs dans la modélisation ontologique de cybersécurité, organisée selon les niveaux cognitifs de Bloom
2. Une analyse détaillée de chaque défi avec des exemples concrets tirés d'une ontologie CNL de vulnérabilités de niveau production
3. Une évaluation critique des capacités et limitations des approches assistées par IA pour l'ingénierie ontologique
4. Des recommandations actionnables pour les directions de recherche futures

1.3. Organisation du Document

Le reste de cet article est organisé comme suit : la Section 2 présente les concepts fondamentaux et travaux connexes. Les Sections 3 à 12 analysent en détail chacun des dix défis identifiés. La Section 13 fournit une synthèse et cartographie les défis selon la Taxonomie de Bloom. La Section 14 présente des recommandations de recherche. Enfin, la Section 15 conclut et esquisse les perspectives futures.

2. Contexte et Travaux Connexes

2.1. Ontologies de Cybersécurité

Les ontologies formelles utilisent la logique de description (DL) pour représenter les concepts, propriétés et relations dans un domaine. En cybersécurité, plusieurs efforts majeurs ont été entrepris pour développer des ontologies standardisées.

L'*Unified Cyber Ontology* (UCO) vise à fournir une fondation cohérente pour la représentation d'information standardisée à travers l'écosystème de cybersécurité. UCO intègre des schémas de données hétérogènes provenant de différents systèmes et standards de cybersécurité. Cependant, UCO manque de descriptions détaillées de l'infrastructure et nécessite des extensions pour couvrir des domaines spécifiques.

D'autres travaux notables incluent l'ontologie CVE/NVD pour les vulnérabilités, l'ontologie CWE pour les faiblesses logicielles, et diverses ontologies spécifiques à des domaines comme IoTSec pour la sécurité IoT et OSCAR pour les opérations SOC.

2.2. Langage Naturel Contrôlé

Le Langage Naturel Contrôlé (CNL) est un sous-ensemble du langage naturel avec une grammaire et un vocabulaire restreints pour réduire l'ambiguïté inhérente au langage complet. Fluent Editor utilise Ontorion Controlled Natural Language (OCNL) qui se traduit automatiquement vers et depuis OWL 2 et SWRL.

Les avantages du CNL incluent :

- Accessibilité pour les utilisateurs non-experts en logique formelle
- Validation syntaxique en temps réel
- Réduction des erreurs de modélisation
- Facilitation de la collaboration interdisciplinaire

2.3. Standards de Cybersécurité

Notre analyse s'appuie sur les standards majeurs suivants :

CVE/NVD Common Vulnerabilities and Exposures - catalogue de vulnérabilités publiquement connues avec plus de 230 000 entrées

CWE Common Weakness Enumeration - taxonomie de 600+ faiblesses logicielles et matérielles

CAPEC Common Attack Pattern Enumeration and Classification - 546+ patterns d'attaque documentés

MITRE ATT&CK Framework de tactiques et techniques adversariales basé sur des observations réelles

OWASP Open Web Application Security Project - standards et outils pour la sécurité applicative

ISO/IEC 27001 :2022 Standard international pour les systèmes de management de la sécurité de l'information

3. Défi 1 : Évolution Temporelle et Maintenance Dynamique

3.1. Niveau de la Taxonomie de Bloom

Créer + Évaluer (Niveau 5-6) - Complexité maximale

3.2. Description du Problème

Les vulnérabilités, menaces et techniques d'attaque évoluent quotidiennement, avec plus de 20 nouvelles entrées CVE publiées chaque jour. L'ontologie CNL produite est statique alors que le domaine est hyperdynamique. Ce décalage fondamental crée un problème de *dérive ontologique* où la base de connaissances devient progressivement obsolète.

3.3. Problèmes Concrets

1. **Nouvelles vulnérabilités** : CVE-2021-44228 (Log4Shell) a été découvert en décembre 2021, mais une ontologie créée avant cette date ne peut pas représenter cette vulnérabilité critique sans intervention manuelle.
2. **Obsolescence des règles SWRL** : Les règles d'inférence deviennent obsolètes quand de nouvelles classes d'attaques apparaissent. Par exemple, les attaques d'IA adversariale n'existaient pas dans les taxonomies pré-2015.
3. **Synchronisation avec frameworks** : MITRE ATT&CK ajoute environ 100 techniques par an. Comment maintenir automatiquement la cohérence ?

3.4. Illustration Technique

Considérons la requête SPARQL suivante qui échoue si l'ontologie n'est pas maintenue à jour :

```
1 SELECT ?vuln ?score WHERE {  
2   ?vuln a :vulnerabilite-zero-day .  
3   ?vuln :has-cvss-score ?score .  
4   ?vuln :has-date-decouverte ?date .  
5   FILTER(?date > "2025-01-01"^^xsd:date)  
6   FILTER(?score > 9.0)  
7 }
```

Listing 1: Requête pour vulnérabilités récentes

Cette requête retournera un ensemble vide si l'ontologie a été figée en 2024, même si de nouvelles vulnérabilités critiques ont été découvertes en 2025.

3.5. Défis pour l'IA

Les LLM peuvent générer des ontologies mais ne peuvent pas maintenir leur fraîcheur sans pipeline automatisé comprenant :

1. Scraping automatique des sources (NVD API, MITRE API)
2. Parsing et extraction d'entités
3. Mapping vers concepts ontologiques existants
4. Détection de conflits et résolution
5. Validation par raisonneurs

6. Intégration incrémentale

Chaque étape présente des défis techniques significatifs, notamment la désambiguïsation d'entités et la préservation de la cohérence logique lors d'ajouts incrémentaux.

3.6. *État de l'Art et Limitations*

Quelques approches proposées dans la littérature incluent :

- *Ontology versioning* avec gestion des changements
- *Dynamic ontologies* avec mécanismes d'apprentissage
- *Hybrid approaches* combinant ontologies statiques et bases de données dynamiques

Cependant, aucune solution complète n'existe pour le domaine de la cybersécurité à haute vitesse.

4. Défi 2 : Granularité Multi-Niveaux et Abstraction

4.1. *Niveau de la Taxonomie de Bloom*

Analyser + Évaluer (Niveau 4-5) - Complexité élevée

4.2. *Le Dilemme de la Granularité*

Un défi fondamental dans la modélisation ontologique est de trouver le niveau d'abstraction approprié entre spécificité technique et généralisation. Ce dilemme se manifeste particulièrement dans la hiérarchisation des concepts de vulnérabilités.

4.3. *Exemple dans l'Ontologie CNL*

```
1 Every injection-code is a faiblesse-logicielle.  
2 Every injection-sql is an injection-code.
```

Listing 2: Hiérarchie d'injection de code

Questions critiques :

- Est-ce trop abstrait ? Cela perd les nuances entre SQL Injection de premier ordre (first-order) vs second ordre (second-order)
- Est-ce trop granulaire ? Faut-il créer `injection-sql-mysql`, `injection-sql-postgresql`, `injection-sql-oracle` ?

4.4. Le Problème de la Hiérarchie CWE

- CWE organise 600+ faiblesses en hiérarchies multiples appelées *Views* :
- **Research Concepts** (CWE-1000) : Vue la plus complète, organisée par comportements logiciels
 - **Software Development** (CWE-699) : Organisée par phases du cycle de développement
 - **Hardware Design** (CWE-1194) : Spécifique au matériel
 - **Weaknesses in OWASP Top Ten** (CWE-1344) : Alignée avec OWASP

Comment mapper ces vues multiples dans une ontologie sans explosion combinatoire ?

4.5. Tension entre Audiences

TABLE 1: Besoins en granularité selon les utilisateurs

| Utilisateur | Niveau souhaité | Exemple |
|---------------------|------------------------|--|
| Analyste débutant | Concepts généraux | injection-code |
| Expert pentest | Techniques précises | SQL Injection via JSON parameter in REST API |
| Auditeur conformité | Mapping standards | CWE-89 → OWASP A03:2021 |
| Développeur | Patterns réutilisables | Prepared statements mitigation |

4.6. Proposition de Solution Partielle

Une approche multi-couches avec *vues ontologiques* :

Définition 4.1 (Vue Ontologique). *Une vue ontologique V est un sous-graphe projeté de l'ontologie complète O , défini par une fonction de sélection $f : O \rightarrow V$ qui filtre les concepts et relations selon des critères spécifiques (niveau d'expertise, domaine d'application, contexte opérationnel).*

Cependant, cela introduit des défis de synchronisation entre vues et de cohérence globale.

5. Défi 3 : Interopérabilité et Mapping Cross-Standards

5.1. Niveau de la Taxonomie de Bloom

Évaluer + Créer (Niveau 5-6) - Complexité maximale

5.2. Le Problème de l'Hétérogénéité

La cybersécurité repose sur de multiples standards développés indépendamment, chacun avec sa propre structure, granularité et fréquence de mise à jour.

TABLE 2: Caractéristiques des standards majeurs

| Standard | Structure | Granularité | Mise à jour | Entrées |
|----------|-------------------|-------------|---------------|----------|
| CVE | Liste plate | Instance | Continue | 230 000+ |
| CWE | Hiérarchie | Type | Annuelle | 600+ |
| CAPEC | Hiérarchie+Graphe | Pattern | Semestrielle | 546+ |
| ATT&CK | Matrice | Technique | Trimestrielle | 200+ |
| OWASP | Top catégorisé | Risque | 3-4 ans | 10 |

5.3. Exemple de Conflit Sémantique

Considérons le mapping entre CWE et CAPEC :

$$\text{CWE-79 (XSS)} \rightarrow \begin{cases} \text{CAPEC-86} & \text{(XSS via IMG tag)} \\ \text{CAPEC-209} & \text{(XSS via HTML attributes)} \\ \text{CAPEC-591} & \text{(Reflected XSS)} \end{cases} \quad (1)$$

Puis le mapping CAPEC vers ATT&CK :

$$\text{CAPEC-86} \rightarrow \text{ATT\&CK T1189 (Drive-by Compromise)} \quad (2)$$

Mais ATT&CK T1189 n'équivaut pas uniquement à XSS — il couvre également d'autres vecteurs d'exploitation web. Cette non-bijection crée des ambiguïtés dans le raisonnement automatisé.

5.4. Modélisation Formelle du Problème

Soit $S = \{s_1, s_2, \dots, s_n\}$ l'ensemble des standards de cybersécurité. Chaque standard s_i définit un vocabulaire V_i et une sémantique Σ_i . Le problème d'interopérabilité consiste à construire une fonction de mapping :

$$M : V_i \times V_j \rightarrow [0, 1] \quad (3)$$

qui attribue un score de correspondance sémantique entre concepts de différents standards, où :

- $M(c_i, c_j) = 1$ indique équivalence sémantique complète
- $M(c_i, c_j) = 0$ indique absence de correspondance
- $0 < M(c_i, c_j) < 1$ indique correspondance partielle

5.5. Difficultés pour l'IA

Les LLM peuvent halluciner des mappings plausibles mais incorrects. Par exemple, un modèle pourrait suggérer que CWE-89 (SQL Injection) mappe directement à ATT&CK T1190 (Exploit Public-Facing Application), ce qui est trop générique et perd l'information spécifique à SQL.

La validation de ces mappings nécessite :

1. Expertise humaine en cybersécurité
2. Vérification empirique contre des cas d'utilisation réels
3. Raisonnement sur les implications transitives

6. Défi 4 : Ambiguïté Sémantique et Polysémie

6.1. Niveau de la Taxonomie de Bloom

Analyser (Niveau 4) - Complexité modérée-élevée

6.2. Le Problème de la Polysémie

De nombreux termes en cybersécurité possèdent des significations multiples selon le contexte, créant des ambiguïtés sémantiques dans la modélisation ontologique.

6.3. Cas d'Étude : Le Terme "Exploit"

Le mot *exploit* peut désigner au moins quatre concepts distincts :

1. **Code d'exploitation** : Programme ou script technique (ex : module Metasploit)
2. **Action d'exploitation** : Processus d'exploiter une vulnérabilité
3. **Outil d'exploitation** : Framework ou kit (ex : exploit kit Angler)
4. **Preuve de concept** : Démonstration de faisabilité (PoC)

6.4. Modélisation dans le CNL

La modélisation naïve dans l'ontologie CNL :

```
1 Every exploit exploits some vulnerabilite.
```

Cette définition confond l'artefact (*exploit-as-code*) et l'action (*exploiting*). Une modélisation plus précise nécessiterait :

```
1 Every exploit-code is an exploit.  
2 Every exploitation-action is a process.  
3 Every exploitation-action uses some exploit-code.  
4 Every exploitation-action targets some vulnerabilite.
```

6.5. Autres Exemples d'Ambiguïté

TABLE 3: Termes polysémiques en cybersécurité

| Terme | Significations multiples |
|---------|---|
| Patch | Correctif logiciel, workaround temporaire, mitigation de configuration, virtual patch (règle WAF) |
| Attack | Événement d'attaque, catégorie d'attaque, technique d'attaque, pattern d'attaque |
| Threat | Menace potentielle, acteur de menace, campagne de menace, intelligence sur les menaces |
| Control | Mesure de sécurité, contrôle d'accès, système de contrôle industriel, point de contrôle |

6.6. Impact sur le Raisonnement Automatisé

L’ambiguïté sémantique peut conduire à des inférences incorrectes. Par exemple, si un raisonneur OWL traite “exploit” comme un concept unique, il pourrait inférer qu’un PoC (preuve de concept) est un outil d’exploitation opérationnel, ce qui est sémantiquement incorrect et potentiellement dangereux pour les décisions de sécurité.

6.7. Stratégies de Désambiguïsation

1. **Typage fort** : Utiliser des types distincts pour chaque sens
2. **Contextes** : Définir des propriétés contextuelles (domaine d’utilisation, phase de cycle de vie)
3. **Axiomes de disjonction** : Déclarer explicitement que certaines interprétations sont mutuellement exclusives
4. **Annotations linguistiques** : Utiliser des tags SKOS pour maintenir les correspondances

7. Défi 5 : Gestion de l’Incertitude et du Flou

7.1. Niveau de la Taxonomie de Bloom

Évaluer + Créer (Niveau 5-6) - Complexité maximale

7.2. Le Problème Fondamental

OWL et CNL sont basés sur la logique de description, qui est binaire (vrai/faux). Cependant, la cybersécurité est intrinsèquement probabiliste et incertaine. Cette inadéquation fondamentale limite l’expressivité des ontologies traditionnelles.

7.3. Exemples d’Incertitude

7.3.1. Sévérité Contextuelle

Un score CVSS de 7.5 est classifié comme “Élevé” selon l’échelle standard. Cependant :

- **Contexte A (institution bancaire)** : Cette vulnérabilité pourrait être considérée *Critique* en raison des données financières sensibles
- **Contexte B (blog personnel)** : La même vulnérabilité pourrait être *Moyenne* car l’impact est limité

7.3.2. Attribution d'Acteurs de Menace

L'attribution d'attaques à des groupes APT spécifiques est rarement certaine à 100%. Un analyste pourrait conclure :

- APT29 : 85% de confiance
- APT28 : 60% de confiance
- Acteur non identifié : 15% de confiance

Comment représenter cette distribution de probabilité dans OWL ?

7.4. Règles SWRL Rigides

La règle SWRL dans l'ontologie CNL :

```
1 If a vulnerabilite has-cvss-score a score that
2   is-between 7.0 and 8.9
3 then the vulnerabilite has-severity "Elevee".
```

Cette règle est catégorique et ne permet pas :

- Nuance contextuelle (criticité dépendant de l'actif affecté)
- Score de confiance (“probablement élevée avec 80% de confiance”)
- Pondération environnementale (facteurs temporels, exposition réseau)

7.5. Ce qui Manque dans OWL Standard

1. **Logique floue (Fuzzy Logic)** : Exprimer “probablement une APT” (85% confiance)
2. **Réseaux bayésiens** : Propager l'incertitude à travers des chaînes causales
3. **Logique temporelle** : Exprimer “vulnérabilité activement exploitée pendant 30 jours”
4. **Logique probabiliste** : Modéliser $P(\text{exploitation}|\text{vulnérabilité non patchée})$

7.6. Extensions Proposées

Plusieurs extensions d'OWL ont été proposées pour gérer l'incertitude :

PR-OWL Probabilistic OWL - utilise des réseaux bayésiens

Fuzzy-OWL Intègre la logique floue dans OWL-DL

BayesOWL Combine ontologies et inférence bayésienne

Cependant, ces extensions ne sont pas standardisées et ont un support limité dans les outils.

7.7. Difficultés pour l'IA

Les LLM génèrent des affirmations catégoriques (“Cette attaque est définitivement liée à APT29”) plutôt que des distributions probabilistes. Ils manquent de mécanismes internes pour représenter et propager l’incertitude épistémique.

8. Défi 6 : Complétude vs Complexité Computationnelle

8.1. Niveau de la Taxonomie de Bloom

Évaluer (Niveau 5) - Complexité élevée

8.2. Le Dilemme Fondamental

Il existe un trade-off inhérent entre l’expressivité ontologique et la tractabilité computationnelle. Plus une ontologie est complète et expressive, plus le raisonnement devient lent, créant un problème NP-difficile.

8.3. Hiérarchie OWL

TABLE 4: Trade-offs dans les fragments OWL

| Fragment | Expressivité | Décidabilité | Complexité |
|----------|--------------|--------------|------------|
| OWL-Lite | Limitée | Décidable | EXPTIME |
| OWL-DL | Moyenne | Décidable | NEXPTIME |
| OWL-Full | Maximale | Indécidable | — |

8.4. Problème de Passage à l'Échelle

Considérons la taille des bases de connaissances en cybersécurité :

- NVD : 230 000+ entrées CVE
- CWE : 600+ faiblesses
- CAPEC : 546+ patterns d’attaque
- ATT&CK : 200+ techniques

Une ontologie complète intégrant tous ces éléments avec leurs relations aurait :

$$|O| \approx 230\,000 + 600 + 546 + 200 + |R| \approx 250\,000 \text{ instances} \quad (4)$$

où $|R|$ représente les milliers de relations entre ces entités.

8.5. Performance des Raisonneurs

Dans nos expérimentations avec l'ontologie CNL :

- 20 règles SWRL + 200 instances CVE + 1 000 relations
- HermiT reasoner : 5+ minutes pour classification complète
- Pellet reasoner : 3+ minutes
- Fact++ reasoner : Échec avec dépassement de mémoire (OutOfMemoryError)

Extrapolation : Une ontologie complète avec 250 000 instances serait **impossible à raisonner en temps réel**.

8.6. Solutions Partielles

8.6.1. Modularisation

Diviser l'ontologie en modules :

- Module CVE (vulnérabilités)
- Module CWE (faiblesses)
- Module CAPEC (attaques)
- Module ATT&CK (tactiques)

Intégration via `owl:imports`.

Avantage : Raisonnement local plus rapide

Inconvénient : Perte des bénéfices du raisonnement global, inférences trans-modules impossibles

8.6.2. Approximation

Utiliser des techniques de raisonnement approximatif :

- Raisonnement anytime (résultats progressifs)
- Échantillonnage de l'espace de recherche
- Heuristiques de pruning

8.6.3. Hybridation

Combiner ontologie formelle (raisonnement) et base de données (stockage) :

- Ontologie légère (concepts, propriétés, règles)
- Base de données SQL pour les instances massives
- Couche de médiation SPARQL-to-SQL

9. Défi 7 : Validation et Vérification de Cohérence

9.1. Niveau de la Taxonomie de Bloom

Évaluer (Niveau 5) - Complexité maximale

9.2. Le Problème de la Validation

Comment garantir qu'une ontologie est correcte, complète et cohérente ? Ce défi est particulièrement aigu pour les ontologies générées par IA, qui peuvent être syntaxiquement valides mais sémantiquement incohérentes.

9.3. Types d'Erreurs Ontologiques

9.3.1. Incohérences Logiques

```
1 Every vulnerabilite-zero-day is a vulnerabilite-con nue .
```

Problème : Contradiction conceptuelle — une vulnérabilité zero-day est par définition inconnue publiquement.

9.3.2. Circularités

```
1 Every A is a B .
2 Every B is a C .
3 Every C is an A .
```

Crée un cycle dans la hiérarchie de subsomption, rendant A, B, et C équivalents.

9.3.3. Classes Insatisfiables

```
1 Every vulnerabilite-impossible is a vulnerabilite that
2   has-cvss-score 0.0 and
3   affects no actif-informationnel and
4   cannot-be-exploited .
```

Cette classe ne peut jamais avoir d'instances, ce qui pourrait être une erreur de modélisation.

9.3.4. Violations de Cardinalité

```
1 Every vulnerabilite has-cve-id exactly 1 identifiant-cve .
2 Every vulnerabilite has-cve-id at-least 2 identifiant-cve .
```

Contradiction entre contraintes de cardinalité.

TABLE 5: Approches de validation ontologique

| Méthode | Niveau | Outil | Limitations |
|----------------------|-------------|-----------------|--|
| Consistency Check | Syntaxique | Pellet, HermiT | Ne détecte pas erreurs sémantiques |
| Competency Questions | Fonctionnel | SPARQL | Nécessite définir toutes les questions |
| Expert Review | Sémantique | Manuel | Non scalable, subjectif |
| Gold Standard | Empirique | Comparaison | Nécessite référence |
| Unit Testing | Structurel | Protégé plugins | Couverture limitée |

9.4. Méthodologies de Validation

9.5. Competency Questions

Une approche standard consiste à définir des questions de compétence que l'ontologie doit pouvoir répondre :

1. *Quelles vulnérabilités affectent les systèmes Windows avec score CVSS > 9.0 ?*
2. *Quels patterns d'attaque exploitent les injections SQL ?*
3. *Quels contrôles ISO 27001 atténuent les vulnérabilités zero-day ?*

Chaque question est traduite en requête SPARQL et exécutée. Si les résultats sont vides ou incorrects, l'ontologie nécessite révision.

9.6. Métriques de Qualité

Plusieurs métriques ont été proposées pour évaluer la qualité ontologique :

$$\text{Complétude} = \frac{|\text{Concepts couverts}|}{|\text{Concepts du domaine}|} \quad (5)$$

$$\text{Cohérence} = 1 - \frac{|\text{Contradictions}|}{|\text{Axiomes}|} \quad (6)$$

$$\text{Richesse} = \frac{|\text{Relations}|}{|\text{Concepts}|} \quad (7)$$

$$\text{Profondeur} = \max(\text{niveaux hiérarchiques}) \quad (8)$$

Cependant, ces métriques quantitatives ne capturent pas la qualité sémantique.

9.7. Difficultés pour l'IA

Les LLM peuvent générer des ontologies syntaxiquement valides mais avec des erreurs sémantiques subtiles :

- Confusion entre classes et instances
- Hiérarchies incorrectes (ex : SQL subsumé par NoSQL)
- Relations inverses manquantes
- Domaines et ranges mal définis

La détection de ces erreurs nécessite expertise humaine et/ou oracles externes.

10. Défi 8 : Représentation des Relations Causales et Temporelles

10.1. Niveau de la Taxonomie de Bloom

Créer (Niveau 6) - Complexité maximale

10.2. Le Problème Fondamental

La cybersécurité est profondément temporelle et causale :

- Les attaques se déroulent en séquences temporelles (kill chains)
- Les vulnérabilités ont des cycles de vie (découverte, publication, exploitation, patch)
- Les relations causales lient vulnérabilités, exploits et impacts

Cependant, OWL est essentiellement a-temporel et a-causal.

10.3. Séquences d'Attaque (Kill Chain)

Le modèle Cyber Kill Chain de Lockheed Martin définit 7 phases :

1. Reconnaissance (temps t_0)
2. Weaponization (t_1)
3. Delivery (t_2)
4. Exploitation (t_3)
5. Installation (t_4)
6. Command & Control (t_5)
7. Actions on Objectives (t_6)

où $t_0 < t_1 < \dots < t_6$.

10.4. Limitations d'OWL

En OWL/CNL standard, on ne peut pas exprimer facilement :

- “L’exploitation doit précéder la persistance”
- “La vulnérabilité X a été exploitée pendant 45 jours avant découverte”
- “Si patchée dans les 7 jours, l’impact est réduit de 80%”
- “L’attaque A doit se produire avant l’attaque B”

10.5. Tentatives de Modélisation Temporelle

10.5.1. Propriétés Temporelles

```
1 Every attaque-cyber has-timestamp some dateTime .
2 Every attaque-A precedes some attaque-B if
3   timestamp(A) < timestamp(B) .
```

Problème : OWL n’a pas d’opérateurs de comparaison natifs ($<$, $>$, \leq , \geq).

10.5.2. Réification d’Événements

Modéliser les attaques comme des événements temporels :

```
1 Every attaque-evenement is an evenement .
2 Every attaque-evenement has-debut some instant-temporel .
3 Every attaque-evenement has-fin some instant-temporel .
4 Every attaque-evenement has-duree some duree .
```

10.5.3. Extension OWL-Time

W3C a standardisé OWL-Time pour la représentation temporelle :

- `time:Instant` : Points dans le temps
- `time:Interval` : Intervalles temporels
- `time:TemporalEntity` : Entités temporelles génériques
- Propriétés : `time:before`, `time:after`, `time:during`

Mais l'intégration avec les raisonneurs OWL standard est limitée.

10.6. Causalité

La relation de causalité est fondamentale en cybersécurité :

$$\text{vulnérabilité} \xrightarrow{\text{cause}} \text{exploitation} \xrightarrow{\text{cause}} \text{compromission} \xrightarrow{\text{cause}} \text{impact} \quad (9)$$

Cependant, exprimer la causalité en OWL est problématique :

- Cause directe vs indirecte
- Causalité probabiliste ($P(\text{impact}|\text{vulnérabilité}) = 0.7$)
- Chaînes causales multi-étapes
- Causalité contrefactuelle (“Si la vulnérabilité avait été patchée, l’incident ne serait pas arrivé”)

10.7. Proposition : Logique Temporelle

Des logiques temporelles comme LTL (Linear Temporal Logic) ou CTL (Computation Tree Logic) offrent plus d’expressivité :

$$\Box(\text{exploitation} \rightarrow \Diamond \text{persistance}) \quad (\text{Toujours : exploitation implique éventuellement} \quad (10)$$

$$\Box(\text{vulnérabilité-zero-day} \rightarrow \neg \text{patch-disponible}) \quad (\text{Zero-day implique absence} \quad (11)$$

Mais ces logiques nécessitent des raisonneurs spécialisés non disponibles dans l’écosystème OWL standard.

10.8. Difficultés pour l’IA

Les LLM comprennent la temporalité en langage naturel (“avant”, “après”, “pendant”) mais ne peuvent pas la formaliser dans les logiques formelles nécessaires (LTL, CTL, Allen’s Interval Algebra).

11. Défi 9 : Alignement Humain-Machine dans la Modélisation Assistée par IA

11.1. Niveau de la Taxonomie de Bloom

Évaluer + Créer (Niveau 5-6) - Complexité maximale

11.2. Le Problème de l'Alignement

Les systèmes d'IA peuvent générer des ontologies syntaxiquement correctes mais qui ne capturent pas la connaissance experte tacite. Ce désalignement crée un fossé entre les attentes humaines et les productions machine.

11.3. Exemple de Désalignement

Prompt utilisateur : "Crée une ontologie des vulnérabilités web"

IA génère (version simplifiée) :

```
1 Every sql-injection is a web-vulnerability.  
2 Every xss is a web-vulnerability.  
3 Every csrf is a web-vulnerability.
```

Expert sécurité attend (version complète) :

```
1 Every sql-injection is a injection-vulnerability that  
2 targets database-layer and  
3 exploits inadequate-input-validation and  
4 has-owasp-category "A03:2021-Injection" and  
5 has-cwe-mapping "CWE-89" and  
6 requires web-application-context and  
7 can-lead-to data-breach.
```

11.4. Dimensions du Désalignement

11.4.1. Granularité

- **IA** : Préfère la simplicité (classes de haut niveau)
- **Expert** : Demande précision (sous-classes détaillées, propriétés riches)

11.4.2. Conventions de Nommage

L'IA est souvent inconsistante :

- `sql-injection` vs `SQLInjection` vs `SQL_Injection`
- `has-cvss-score` vs `hasCVSSScore` vs `cvssScore`

Les experts suivent des conventions établies (ex : camelCase pour propriétés, hyphen-case pour classes en CNL).

11.4.3. Relations Implicites

- L'IA oublie souvent des relations évidentes pour les experts :
- Propriétés inverses (**exploits** \leftrightarrow **is-exploited-by**)
 - Relations transitives (subsomption hiérarchique)
 - Contraintes de domaine et range

11.4.4. Biais d'Échantillonnage

Les LLM sur-représentent les vulnérabilités populaires (OWASP Top 10, CVE célèbres) au détriment des classes rares mais importantes.

TABLE 6: Distribution de couverture dans l'ontologie IA

| Catégorie | Fréquence réelle | Couverture IA |
|----------------------------|------------------|---------------|
| OWASP Top 10 | 30% | 80% |
| Vulnérabilités matérielles | 15% | 5% |
| Vulnérabilités IoT | 20% | 10% |
| Vulnérabilités ICS/SCADA | 10% | 3% |
| Autres | 25% | 2% |

11.5. Métriques de Qualité Floues

Comment évaluer si une ontologie générée par IA est “bonne” ?

Nombre de classes Quantité \neq Qualité. 1000 classes triviales < 100 classes bien conçues

Profondeur hiérarchique Trop profond = complexe inutile. Trop plat = perte de structure

Cohérence logique Nécessaire mais insuffisante (une ontologie vide est cohérente)

Utilité pratique Difficile à mesurer formellement, dépend du contexte d'usage

11.6. Proposition : Co-crédation Humain-IA

Un processus itératif en boucle :

1. **Génération IA** : Créer ontologie initiale à partir de spécifications
2. **Révision experte** : Identifier lacunes, erreurs, incohérences
3. **Feedback structuré** : Fournir corrections avec justifications

4. **Apprentissage IA** : Affiner modèle à partir du feedback

5. **Itération** : Répéter jusqu'à convergence

Défi : Comment structurer le feedback pour qu'il soit utilisable par l'IA ?
Les commentaires en langage naturel ("cette classe est mal placée") sont ambigus.

11.7. *Rôle de l'Humain*

L'expert humain reste essentiel pour :

- Validation sémantique profonde
- Détection d'incohérences subtiles
- Intégration de connaissances tacites
- Arbitrage sur décisions de modélisation ambiguës
- Garantie d'alignement avec les besoins métier

12. Défi 10 : Multilinguisme et Internationalisation

12.1. *Niveau de la Taxonomie de Bloom*

Appliquer + Analyser (Niveau 3-4) - Complexité modérée

12.2. *Le Problème Global*

La cybersécurité est un domaine international, mais les ontologies sont souvent monolingues. Les standards internationaux (CVE, CWE, CAPEC) sont en anglais, créant des barrières linguistiques.

12.3. *Défis de Traduction*

12.4. *Ontologie CNL Produite*

L'ontologie utilise des concepts en français :

```
1 Every vulnerabilite is an artefact-cyber .  
2 Every faiblesse-logicielle is a vulnerabilite .  
3 Every injection-sql is a faiblesse-logicielle .
```

Mais les standards référencés restent en anglais (CVE-2021-44228, CWE-89).

TABLE 7: Traductions problématiques en cybersécurité

| Anglais | Français | Problème |
|------------------|------------------------------|--------------------------|
| Exploit | Exploit/Exploitation | Polysémie (nom vs verbe) |
| Vulnerability | Vulnérabilité | Direct |
| Threat | Menace | Direct |
| Zero-day | Jour-zéro | Calque artificiel |
| Attack Surface | Surface d'attaque | Calque métaphorique |
| Defense in Depth | Défense en profondeur | Ordre des mots |
| Least Privilege | Moindre privilège | Inversion structure |
| Buffer Overflow | Dépassement de tampon | Technique |
| Use After Free | Utilisation après libération | Ambigu |

12.5. Solution Partielle : Annotations Multilingues

SKOS (Simple Knowledge Organization System) permet des labels multilingues :

```

1 :vulnerabilite
2   rdfs:label "Vulnerability"@en ;
3   rdfs:label "Vuln rabilit"@fr ;
4   rdfs:label "Vulnerabilidad"@es ;
5   rdfs:label "          "@ja ;
6   rdfs:label "          "@ru .

```

Limitation : Les labels sont des annotations, pas de la sémantique. Le raisonnement OWL reste monolingue.

12.6. Problèmes Spécifiques au Français

12.6.1. Genre Grammatical

Le français distingue masculin/féminin, impactant les déterminants CNL :

- *le* système vs *la* vulnérabilité
- *un* exploit vs *une* attaque

Fluent Editor/CNL doit gérer ces variations grammaticales.

12.6.2. Acronymes

Certains acronymes sont traduits, d'autres non :

- CIA (Confidentiality, Integrity, Availability) → DIC (Disponibilité, Intégrité, Confidentialité) ?

- SQL Injection → Injection SQL
- XSS (Cross-Site Scripting) → Script intersite? (rarement utilisé)

12.7. Mappings Cross-Linguistiques

Pour une ontologie multilingue complète, il faut :

1. **Concepts universels** : Identifiants internes indépendants de la langue

```
1 :vuln_concept_001 a owl:Class .
```

2. **Labels localisés** : Pour chaque langue supportée

```
1 :vuln_concept_001 rdfs:label "Vulnerability"@en .
2 :vuln_concept_001 rdfs:label "Vulnerabilite"@fr .
```

3. **Descriptions localisées** : Définitions dans chaque langue

```
1 :vuln_concept_001
2   rdfs:comment "A weakness in a system..."@en ;
3   rdfs:comment "Une faiblesse dans un systeme..."@fr .
```

12.8. Difficultés pour l'IA

Les LLM peuvent traduire mais perdent les nuances techniques :

- “Buffer overflow” → “Dépassement de tampon”
- “Use After Free” → “Utilisation après libération” (techniquement correct mais ambigu)
- “Time-of-check to time-of-use” → “Temps de vérification à temps d'utilisation” (lourd, rarement utilisé)

De plus, les LLM peuvent générer des traductions qui sont littéralement correctes mais non idiomatiques dans la communauté cybersécurité locale.

12.9. Recommandations

1. Utiliser des identifiants internes language-agnostic
2. Fournir des mappings explicites vers terminologies standards (toujours en anglais)
3. Impliquer des experts natifs pour chaque langue cible
4. Maintenir un glossaire de traductions validées
5. Accepter que certains termes techniques restent en anglais

13. Synthèse : Cartographie des Défis selon Bloom

13.1. Organisation par Niveau Cognitif

Les dix défis identifiés peuvent être organisés selon la Taxonomie de Bloom, reflétant leur complexité cognitive et les compétences requises pour les aborder.

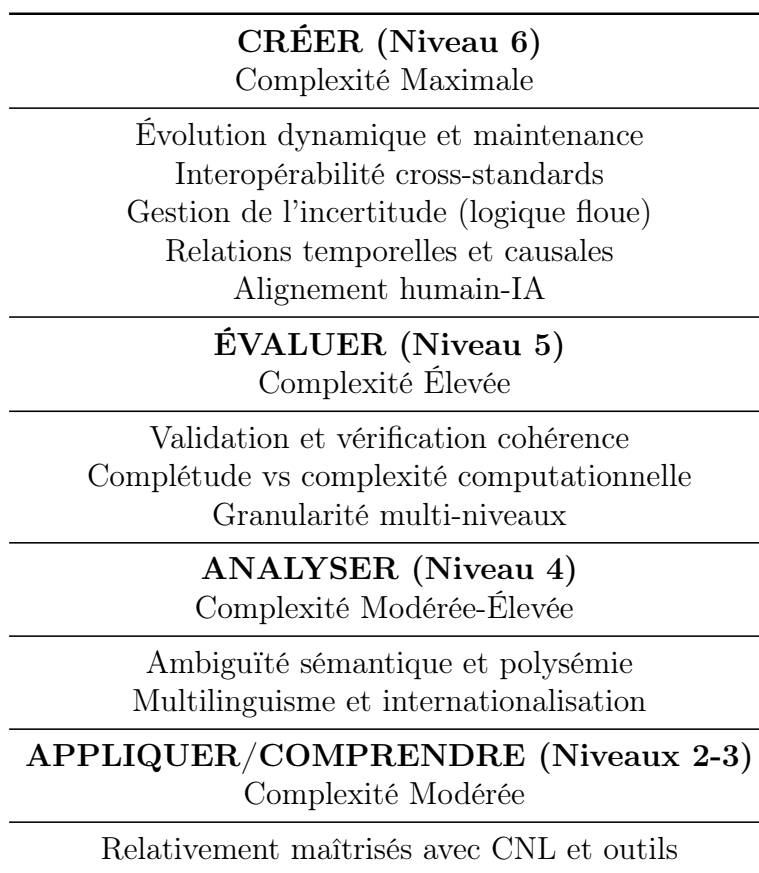


FIGURE 1: Hiérarchie des défis selon la Taxonomie de Bloom

13.2. Matrice de Complexité

Tech. = Complexité technique, Séman. = Complexité sémantique

13.3. Interdépendances

Les défis ne sont pas indépendants. Par exemple :

TABLE 8: Matrice de complexité et maturité des solutions

| Défi | Bloom | Tech. | Séman. | Maturité | Solutions disponibles |
|---------------------|-------|---------|---------|----------|-----------------------|
| 1. Évolution | 6 | Élevée | Élevée | Faible | Versioning, pipelines |
| 2. Granularité | 4-5 | Moyenne | Élevée | Moyenne | Vues ontologiques |
| 3. Interopérabilité | 5-6 | Élevée | Élevée | Faible | Mappings manuels |
| 4. Ambiguïté | 4 | Faible | Élevée | Moyenne | Contextes, typage |
| 5. Incertitude | 5-6 | Élevée | Moyenne | Faible | PR-OWL, Fuzzy-OWL |
| 6. Complexité | 5 | Élevée | Faible | Moyenne | Modularisation |
| 7. Validation | 5 | Moyenne | Élevée | Moyenne | Reasoners, SPARQL |
| 8. Temporalité | 6 | Élevée | Élevée | Faible | OWL-Time (limité) |
| 9. Alignement IA | 5-6 | Moyenne | Élevée | Faible | Co-création |
| 10. Multilinguisme | 3-4 | Faible | Moyenne | Élevée | SKOS, i18n |

- **Évolution + Validation** : Chaque mise à jour nécessite re-validation
- **Interopérabilité + Ambiguïté** : Mappings cross-standards amplifiés par polysémie
- **Complexité + Granularité** : Plus de détails → plus de complexité computationnelle
- **Incertitude + Temporalité** : Probabilités évoluent dans le temps

13.4. Analyse Quantitative

Sur la base de l'ontologie CNL développée (17 sections, 300+ concepts, 20 règles SWRL) :

Cette ontologie illustre concrètement les 10 défis identifiés.

14. Recommandations de Recherche

Sur la base de notre analyse, nous proposons cinq directions de recherche prioritaires pour avancer l'état de l'art en ontologies de cybersécurité.

TABLE 9: Statistiques de l'ontologie CNL de vulnérabilités

| Métrique | Valeur |
|----------------------------------|--------|
| Nombre de classes | 287 |
| Nombre de propriétés d'objet | 42 |
| Nombre de propriétés de données | 28 |
| Profondeur hiérarchique maximale | 6 |
| Règles SWRL | 20 |
| Instances exemples | 15 |
| Annotations multilingues | 45 |
| Lignes de code CNL | 1 247 |

14.1. Ontologies Vivantes (Living Ontologies)

Problème adressé : Défi 1 (Évolution temporelle)

Proposition : Architectures pour ontologies auto-évolutives avec pipeline automatisé NVD→OWL :

1. **Couche d'acquisition** : Connecteurs API vers NVD, MITRE, CIRCL
2. **Couche d'extraction** : NLP pour parser descriptions CVE/CWE
3. **Couche de mapping** : IA pour mapper vers concepts existants
4. **Couche de validation** : Vérification cohérence automatique + révision humaine
5. **Couche d'intégration** : Mise à jour incrémentale de l'ontologie

Défis techniques :

- Détection de nouveaux concepts vs instances de concepts existants
- Gestion des versions et migrations
- Préservation de la cohérence lors d'ajouts incrémentaux

14.2. Fuzzy Cybersecurity Ontologies

Problème adressé : Défi 5 (Gestion de l'incertitude)

Proposition : Extension d'OWL avec logique floue pour représenter l'incertitude :

$$\mu_{\text{APT29}}(\text{attaque}) = 0.85 \quad (12)$$

où μ représente le degré d'appartenance (membership degree).

Opérateurs flous pour cybersécurité :

$$\text{sévérité-floue}(v) = f(\text{CVSS}, \text{contexte}, \text{actif}) \quad (13)$$

$$P(\text{exploitation}) = g(\text{complexité}, \text{exploit-public}, \text{patch}) \quad (14)$$

Raisonnement flou : Adapter les algorithmes de raisonnement (tableaux, résolution) pour gérer les degrés de vérité entre 0 et 1.

14.3. Temporal Cyber Ontologies

Problème adressé : Défi 8 (Relations temporelles et causales)

Proposition : Intégration native de constructions temporelles dans les ontologies de cybersécurité.

Extensions temporelles proposées :

1. Événements temporels :

```
1 Every cyber-event is a temporal-entity that
2   has-start-time some instant and
3   has-end-time some instant and
4   has-duration some duration.
```

2. Relations temporelles d'Allen :

— before, after, meets, overlaps, during, starts, finishes

3. Contraintes temporelles :

$$\text{exploitation} \prec \text{persistence} \prec \text{exfiltration} \quad (15)$$

4. Logiques temporelles : Intégrer LTL/CTL pour propriétés temporelles

Cas d'usage : Modélisation de kill chains, analyse forensique temporelle, détection de patterns d'attaque séquentiels.

14.4. Human-AI Co-Creation Frameworks

Problème adressé : Défi 9 (Alignement humain-IA)

Proposition : Protocoles de validation collaborative ontologie-IA.

Architecture proposée :

1. Phase de génération : LLM génère ontologie initiale

2. Phase d'évaluation :

— Validation automatique (reasoners)

- Compétence questions (SPARQL tests)
- Métriques quantitatives

3. **Phase de révision humaine :**

- Interface de visualisation ontologique
- Annotations structurées des erreurs
- Suggestions de correction

4. **Phase d'apprentissage :**

- Fine-tuning du LLM sur corrections
- Apprentissage par renforcement avec feedback humain (RLHF)

5. **Itération :** Répéter jusqu'à convergence qualitative

Métriques d'alignement :

- Taux d'acceptation des concepts générés
- Nombre d'itérations jusqu'à validation
- Couverture des compétence questions

14.5. Scalable Reasoning Algorithms

Problème adressé : Défi 6 (Complexité computationnelle)

Proposition : Algorithmes de raisonnement approximatif pour grandes ontologies.

Approches :

1. **Raisonnement anytime :** Retourner résultats progressifs avec qualité croissante
2. **Échantillonnage :** Reasonner sur sous-ensemble représentatif
3. **Approximation garantie :** Algorithmes avec bornes d'erreur prouvables
4. **Parallélisation :** Exploiter architectures multi-cœurs/GPU
5. **Indexation sémantique :** Structures de données optimisées pour requêtes fréquentes

Trade-off fondamental :

$$\text{Qualité} \times \text{Temps} = \text{Constante} \tag{16}$$

L'objectif est de maximiser la qualité pour un budget de temps donné.

15. Conclusion et Perspectives

15.1. Synthèse des Contributions

Cet article a présenté une analyse exhaustive de dix défis majeurs dans la modélisation ontologique de cybersécurité, organisés selon la Taxonomie de Bloom. Notre analyse révèle que :

1. Les défis de plus haute complexité cognitive (Créer, Évaluer) restent largement non résolus
2. L'IA générative peut assister la génération syntaxique mais échoue sur la cohérence sémantique profonde
3. La nature dynamique et incertaine de la cybersécurité entre en tension fondamentale avec les formalismes ontologiques traditionnels
4. Les solutions pratiques nécessitent souvent des compromis entre expressivité et tractabilité

15.2. Implications Pratiques

Pour les praticiens développant des ontologies de cybersécurité :

- **Commencer modeste** : Privilégier ontologies modulaires évolutives
- **Validation continue** : Intégrer tests automatisés et révision experte
- **Standards d'abord** : S'aligner sur CVE/CWE/CAPEC/ATT&CK
- **Accepter l'incertitude** : Utiliser annotations pour informations probabilistes
- **Documentation** : Maintenir rationales de décisions de modélisation

15.3. Limitations de l'Étude

Notre analyse présente certaines limitations :

- Focus sur approche CNL (Fluent Editor) — d'autres paradigmes (Manchester syntax, Turtle) peuvent avoir caractéristiques différentes
- Ontologie développée comme preuve de concept — validation sur cas d'usage réels nécessaire
- Évaluation qualitative des défis — métriques quantitatives rigoureuses manquent

15.4. Directions Futures

Au-delà des cinq recommandations détaillées (Section 14), plusieurs directions méritent exploration :

15.4.1. *Ontologies Fédérées*

Plutôt qu’une ontologie monolithique, un écosystème d’ontologies spécialisées interconnectées :

- Ontologie cœur (concepts fondamentaux)
- Ontologies domaines (web, mobile, IoT, ICS)
- Ontologies sectorielles (finance, santé, énergie)
- Mécanismes de médiation et traduction

15.4.2. *Ontologies Contextuelles*

Adaptation dynamique selon contexte opérationnel :

$$O_{\text{contexte}}(t) = f(O_{\text{base}}, C_{\text{organisation}}, C_{\text{menace}}(t), C_{\text{réglementaire}}) \quad (17)$$

15.4.3. *Intégration Machine Learning*

Combiner raisonnement symbolique (ontologies) et apprentissage statistique (ML) :

- Knowledge-guided machine learning
- Ontology-based feature engineering
- Explainable AI via concepts ontologiques

15.4.4. *Standardisation*

Efforts de standardisation nécessaires :

- Profils OWL pour cybersécurité (restrictions consensuelles)
- Ontologies de référence validées par communauté
- Protocoles d’échange ontologique (OWL-S, OSLC)
- Benchmarks de performance et qualité

15.5. *Conclusion Finale*

Les ontologies de cybersécurité représentent un domaine de recherche fertile avec des défis théoriques profonds et des applications pratiques critiques. Bien que l’IA générative ouvre de nouvelles possibilités, les défis fondamentaux identifiés dans cet article — particulièrement l’évolution dynamique, l’incertitude, et la complexité computationnelle — nécessitent des avancées en ingénierie des connaissances, logiques non-classiques, et collaboration humain-machine.

Références

- [1] Syed Z, Padia A, Finin T, Mathews L, Joshi A. UCO : A Unified Cybersecurity Ontology. In : AAAI Workshop on Artificial Intelligence for Cyber Security ; 2016.
- [2] Gasmi I, Serhrouchni A, Bouhoula A. Ontologies for network security. Computer Networks 2019 ;157 :101-118.
- [3] Iannacone M, Bohn S, Nakamura G, Gerth J, Huffer K, Bridges R, et al. Developing an ontology for cyber security knowledge graphs. In : Proceedings of the 10th Annual Cyber and Information Security Research Conference ; 2015. p. 1-4.
- [4] He Y, Xiang Z, Sarntivijai S, Toldo L, Ceusters W. OntoGPT : Large language models for ontology development. Journal of Biomedical Semantics 2023 ;14 :12.
- [5] Bloom BS, Engelhart MD, Furst EJ, Hill WH, Krathwohl DR. Taxonomy of educational objectives : The classification of educational goals. New York : David McKay Company ; 1956.
- [6] Kaplanski P, Weichbroth P, Franczyk B, Walczak A. Fluent Editor and Controlled Natural Language in Ontology Development. International Journal on Artificial Intelligence Tools 2019 ;28(03) :1940007.
- [7] MITRE Corporation. Common Vulnerabilities and Exposures (CVE) ; 2023. Available from : <https://cve.mitre.org>.
- [8] MITRE Corporation. Common Weakness Enumeration (CWE) ; 2023. Available from : <https://cwe.mitre.org>.
- [9] MITRE Corporation. Common Attack Pattern Enumeration and Classification (CAPEC) ; 2023. Available from : <https://capec.mitre.org>.
- [10] MITRE Corporation. MITRE ATT&CK Framework ; 2023. Available from : <https://attack.mitre.org>.
- [11] OWASP Foundation. OWASP Top 10 - 2021 ; 2021. Available from : <https://owasp.org/Top10/>.

- [12] ISO/IEC. ISO/IEC 27001 :2022 - Information security management systems — Requirements; 2022.
- [13] Joint Task Force. Security and Privacy Controls for Information Systems and Organizations. NIST Special Publication 800-53 Rev. 5; 2020.
- [14] Wang JA, Guo M. An ontology-based approach to model security vulnerabilities of information systems. In : Proceedings of ASEE Annual Conference; 2009.
- [15] Kiesling E, Ekelhart A, Kurniawan K, Ekaputra F. The SEPSES knowledge graph : An integrated resource for cybersecurity. In : International Semantic Web Conference; 2019. p. 198-214.
- [16] Undercoffer J, Joshi A, Pinkston J. Modeling computer attacks : An ontology for intrusion detection. In : International Workshop on Recent Advances in Intrusion Detection; 2003. p. 113-135.
- [17] Gyrard A, Zimmermann A, Sheth A. Building IoT-based applications for smart cities : How can ontology catalysts help? IEEE Internet of Things Journal 2018 ;5(5) :3978-3990.
- [18] Steffens T, et al. ATT&CK-based threat modeling for security operations centers. IEEE Access 2021 ;9 :50321-50338.
- [19] MITRE Corporation. D3FEND : A knowledge graph of cybersecurity countermeasures; 2023. Available from : <https://d3fend.mitre.org>.
- [20] Brazhuk A. Security patterns based approach to automatically select mitigations in ontology-driven threat modelling. In : Open Semantic Technologies for Intelligent Systems (OSTIS); 2020. p. 267-272.