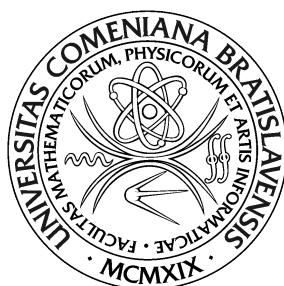


UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY



SÉMANTICKÉ PUBLIKOVANIE SPRAVODAJSKÝCH DÁT

Diplomová práca

2021

Bc. Matej Rychtárik

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY



SÉMANTICKÉ PUBLIKOVANIE SPRAVODAJSKÝCH DÁT

Diplomová práca

Študijný program: Aplikovaná informatika
Študijný odbor: 2511 Aplikovaná informatika
Školiace pracovisko: Katedra aplikovanej informatiky
Školiteľ: doc. RNDr. Martin Homola, PhD.

Bratislava, 2021

Bc. Matej Rychtárik

Čestne prehlasujem, že túto diplomovú prácu som
vypracoval samostatne len s použitím uvedenej literatúry
a za pomoci konzultácií u môjho školiteľa.

Bratislava, 2021

.....

Bc. Matej Rychtárik

Pod'akovanie

Touto cestou by som sa chcel v prvom rade poďakovať môjmu školiteľovi doc. RNDr. Martinovi Homolovi, PhD. za jeho cenné rady a usmernenia, ktoré mi veľmi pomohli pri riešení tejto diplomovej práce.

Abstrakt

Abstract

Obsah

1	Úvod	1
I	Prehľad problematiky	2
2	Sémantický web	3
2.1	Linked Data	5
2.2	Resource Description Framework (RDF)	6
2.3	SPARQL	8
3	Ontológie	10
3.1	Základné pojmy	11
3.2	Využitie ontológií	12
3.3	Syntax ontológií	12
3.3.1	Web Ontology Language	13
3.4	Deskripčná logika	13
4	Existujúce ontologické riešenia	14
4.1	CTI model	15
4.2	Integrated Cyber Analysis System	15
4.3	STUCCO	17

<i>OBSAH</i>	viii
II Návrh riešenia	18
5 Unified Cybersecurity Ontology	19
5.1 Popis ontológie	19
5.2 Model	21

Kapitola 1

Úvod

Nejaky strucny uvod do problematiky

Časť I

Prehľad problematiky

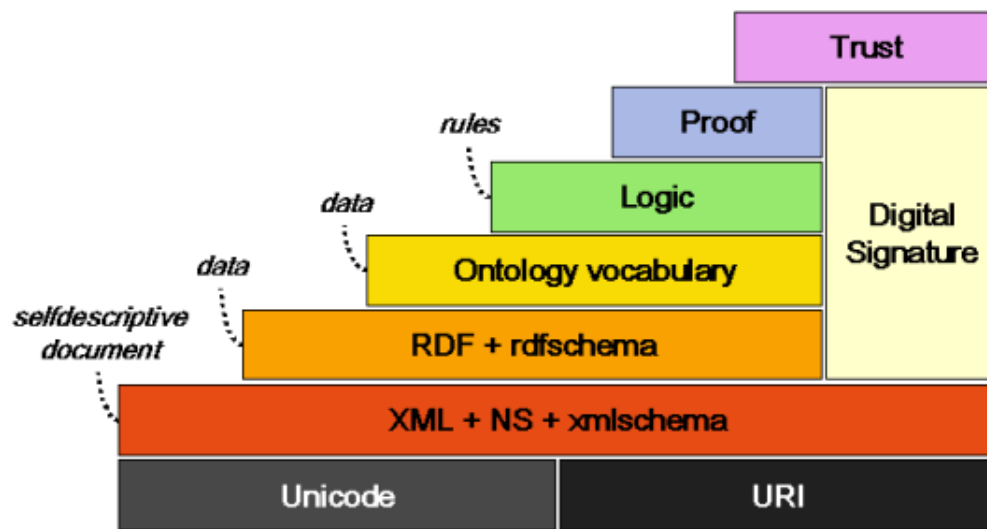
Kapitola 2

Sémantický web

Sémantický web [BLHL01] poskytuje spoločný framework, ktorý umožňuje zdieľanie a opätovné použitie údajov v rámci aplikácií. Štandardy podporujú spoločné dátové formáty a protokoly, kde najpodstatnejším je Resource Description Framework (RDF). Prvýkrát pojem Sémantický web zaviedol Tim Berners-Lee a popisoval "dátový web", ktorý môže byť strojovo čitateľný. Zámerom je zlepšiť prístupnosť informácií publikovaných na webe pre strojové spracovanie. Sémantický web má vrstvovú štruktúru ako si môžeme všimnúť na obrázku 2.1. Jednotlivé údaje sú potrebné až vo vyšších vrstvách.

XML vrstva zaručuje, že môžeme spájať Sémantický web s inými normami, založenými napríklad na XML, ktorá je rozšírená a podporovaná a RDF dáta sa v nej dajú dobre prenášať, spracovávať a uchovávať. Toto je už ale pre dnešné časy neštandardné a viac sa využíva výmena dát pomocou JSON formátu.

RDF je metóda popisovania vecí pomocou vzťahu medzi dvoma objektami. Napríklad koncepčne povedané "Jožko má jablko" je zadaním spojenia medzi objektami "Jožko" a "jablko" pomocou vzťahu "má". Toto spojenie je známe ako triplet, ktorý je základnou stavebnou jednotkou sémantického



Obr. 2.1: Semantic Web - vrstvy.
Zdroj: [KM]

webu.

RDF je aj názov slovníka, ktorý obsahuje množinu preddefinovaných termínov. Tieto termíny sú všeobecne používané na popis dát. Napríklad obsahuje najzákladnejšiu vlastnosť pre objekty a to vlastnosť typu objektu – *rdf:type*.

RDFS je taktiež slovník. V RDF slovníku máme termíny, ktoré nám pomáhajú určovať definície a popisy jednotlivých objektov. V RDFS získavame možnosť popisovať triedy. Pokiaľ si zoberieme triedu "Osoba" a triedu "Žena", vieme pomocou RDFS slovníka zdefinovať vzťah podtriedy vlastnosťou "rdfs:subClassOf". Vďaka slovníkom RDF a RDFS môžeme tvoriť detailné popísanie našich dát.

Ontológia je v našom ponímaní synonymom k slovu slovník. Slovník RDFS môže byť použitý na tvorbu vlastnej ontológie. Sú v nej definované

Vďaka takejto reprezentácii dát je možné písať pravidlá, ktoré má daný súbor dát spĺňať. Vďaka logickej vrstve vieme napríklad zdefinovať pravidlá

ako: Všetky objekty typu "Muž" a "Žena" sú zároveň typom "Osoba" alebo Množina objektov s typom "Muž" je disjunktná s množinou objektov "Žena". Tieto pravidlá slúžia na kontrolu konzistentnosti našich dát.

Text uvedený nižšie popisuje niekoľko technológií, ktoré sú potrebné pre tvorbu sémantického webu.

2.1 Linked Data

Linked Data [BHBL11] je metóda zverjňovania štrukturovaných dát. Ich hlavným cieľom je poprepájať existujúce databázy (primárne písané v RDF formáte), medzi rôznymi údajmi a umožniť ľuďom zdieľať štrukturované dáta na webe pomocou HTML. Časť vízie do budúcnosti je, aby sa Internet stal globálnou databázou. Princípy Linked Data prvýkrát načrtol Tim Berners-Lee. Popísal 4 pravidlá pre zverejňovanie dát na webe:

1. používať URI ako názvy objektov, ktoré sú identifikátormi informácie, jej umiestnenia a ďalších vlastností,
2. používať HTTP URI, aby si ich ľudia vedeli pozrieť,
3. uvádzať informácie o tom, čo názov identifikuje pri vyhľadávaní pomocou otvorených štandardov, ako sú napríklad RDF alebo SPARQL,
4. pri publikovaní údajov na webe, zahrnúť odkazy aj na iné URI, aby sa dalo objavovať viac vecí.

Sú známe aj ako Linked Data princípy.

2.2 Resource Description Framework (RDF)

RDF [SRb] je štandardný model na zakódovanie metadát a ďalších informácií. Je to taktiež formát, ktorý bol navrhnutý a štandardizovaný na reprezentáciu dát pre sémantický web. Zdroje týchto dát sú väčšinou webové zdroje, ktoré môžu byť čokoľvek, napríklad dokumenty, ľudia, fyzické objekty, atď. Taktiež poskytuje spoločný framework na vyjadrenie informácií a možnosť zdieľať ich medzi softvérmí, bez straty ich hodnoty. Dáta sa uchovávajú v Triple Store databázach, ktorých formát je striktne daný. Výhodou je, že dáta môžu byť spracované aj softvérmí, pre ktoré dané dáta neboli vytvorené.

RDF súbor je taký dokument, ktorý ukladá RDF grafy do špecifického formátu serializácie pre RDF, ako sú napríklad N-Triple, Turtle, RDF/XML a mnohé ďalšie. RDF bol postavený na myšlienke vytvárať údaje vo forme predmet-predikát-objekt, ktorý sa volá triplet. Triplet je základná stavebná jednotka akejkoľvek množiny dát zapísaných v RDF. Tieto údaje sú reprezentované ako orientované grafy. Predmet a objekt predstavujú vrcholy a predikát je orientovaná hrana medzi nimi. Predmet môže byť použitý aj ako objekt v inom triplete. Týmto spôsobom sa triplety prepájajú a vzniká z nich grafová databáza. Predmet je vždy definovaný ako URI a popisuje zdroj informácie. Objekt môže byť taktiež nejaké URI popisujúce zdroj, ale taktiež to môže byť primitívna hodnota, ako napríklad string, integer, date, atď. Predikát popisuje, aký vzťah alebo rola medzi predmetom a objektom existuje. Predikát je vždy reprezentovaný ako URI, ktoré pochádza z ontológií (kolekcie viacerých URI).

Na uľahčenie ukladania a čitateľnosti dát sa využívajú takzvané prefixy, ktoré sú preddefinovaním základných URI, do ktorých sa dodáva zvyšná hodnota URI pomocou dvojbodky, ako je to uvedené v nasledujúcom príklade a graficky znázornené v obrázku 2.2.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns> .
```

```
@prefix dbr: <http://dbpedia.org/resource/> .
```

```
@prefix dbo: <http://dbpedia.org/ontology/> .
```

```
@prefix dbp: <http://dbpedia.org/property/> .
```

```
dbr:Bratislava dbo:highestPlace dbr:Devínska_Kobyla .
```

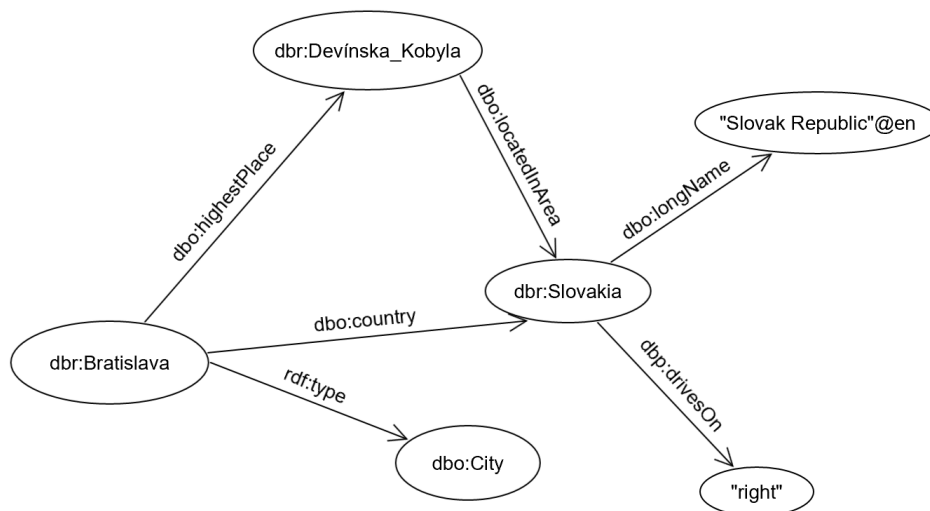
```
dbr:Bratislava rdf:type dbo:City .
```

```
dbr:Bratislava dbo:country dbr:Slovakia .
```

```
dbr:Devínska_Kobyla dbo:locatedInArea dbr:Slovakia .
```

```
dbr:Slovakia dbp:drivesOn "right" .
```

```
dbr:Slovakia dbo:longName "Slovak Republic"@en .
```



Obr. 2.2: Príklad grafovej databázy.

2.3 SPARQL

SPARQL [SRa] je dopytovací jazyk pre RDF databázy, ktorý umožňuje získavanie a manipuláciu s databázou. Bol vytvorený skupinou DAWG, ktorá je súčasťou W3C a je uznávaný ako kľúčová technológia sémantického webu.

Ak by sme porovnali SPARQL s dopytovacím jazykom pre relačné databázy, napr. SQL, zistíme, že sú si podobné v kľúčových slovách, ako sú napr. SELECT, WHERE, FROM atď. SPARQL dopyt využíva trojice ako základný prvok, kde predmet, predikát alebo objekt môžu byť premenné. Dopyt sa robí nad dátovou kolekciou RDF, čo je množina dokumentov, patriaca pod určitý koncový bod - '*endpoint*'. Je to dopytovací jazyk, ktorý z orientovaného ohodnoteného grafu zisťuje hodnoty jednotlivých vrcholov a hrán, ktoré sú výstupnými parametrami dopytu.

```
@prefix dbr: <http://dbpedia.org/resource/> .
```

```
SELECT ?predicate ?object WHERE {
    dbr:Bratislava ?predicate ?object .
}
```

```
+-----+-----+
| ?predicate      | ?object          |
+-----+-----+
| dbo:highestPlace | dbr:Devínska_Kobyla |
| rdf:type         | dbo:City          |
| dbo:country      | dbr:Slovakia      |
+-----+-----+
```

Príklad dopytu nad databázou uvedenou vyššie, spúšťame nad endpointom DBPedia a výsledok je len zlomkom z toho, čo nám skutočne vráti: Chceme získať všetky údaje o Bratislave.

Okrem operácie SELECT poznáme aj ďalšie typy dopytov. ASK je dopyt, ktorý nám vracia pravdivostnú hodnotu pre daný dopyt. Vieme ním napríklad zistiť či sa v našom grafe nachádza mesto Bratislava. Taktiež poznáme dopyt DESCRIBE, ktorý vracia RDF graf opisujúci jednotlivé vlastnosti výsledných hodnôt dopytu. Ako posledný typ dopytu je CONSTRUCT, ktorý vracia nový RDF graf podľa predlohy vytvorenej v hlave dopytu.

Kapitola 3

Ontológia

Výraz ontológia [GOS09] pochádza z gréckeho slova kde '*ontos*' znamená existencia a '*logos*' znamená veda. Ontológia v informatike je uceleným popisom pojmov v určitej oblasti záujmu. Obsahuje určitú klasifikáciu údajov do hierarchicky usporiadaných kategórií a množinu odvodzovacích pravidiel, pomocou ktorých je možné z faktov odvodiť nové skutočnosti. Prostredníctvom ontológií je možné vytvárať spojenia, vykonávať analýzu údajov a sprostredkovať výhody webu obohateného o sémantiku.

Jej cieľmi je zadefinovanie a zdieľanie jednotného zápisu informácií pre danú doménu. Ak napríklad viac stránok využíva na popis pojmov takúto zadefinovanú ontológiu, vedia boti získať a vyhľadávať viac dát o hľadanej informácii.

Taktiež je jej cieľom opätovné použitie ontológie, napríklad ak máme dobre zadefinovanú ontológiu, môžu ontologický inžinieri doplniť do našej ontológie ďalšie vlastnosti a tým by základ ontológie bol rovnaký ale bol by rozšírený o určité dáta, podľa potreby ontologických inžinierov.

3.1 Základné pojmy

Ontológia sa skladá zo základných stavebných prvkov *Trieda*, *Entita*, *Atribút*, *Vzťah*.

Triedy alebo typy definujú skupiny alebo množiny objektov. Triedy majú hierarchickú štruktúru zloženú z ich podtried. Každá podtrieba spĺňa vlastnosti nadtrieby a môže byť rozšírená o vlastné vlastnosti.

Entity sú individuálne inštancie nejakej nami zadanovej triedy. Ak by sme mali entitu *Bratislava*, a triedy *Mesto* a *Hlavné mesto*, kde *Mesto* je podtriedou *Hlavné mesto*, tak nám z ontológie vyplýva, že ak je entita *Bratislava* individuálnou inštanciou triedy *Hlavné mesto*, tak je aj individuálnou inštanciou triedy *Mesto*.

Atribúty sú vlastnosti *Tried* a *Entít* a môžu niesť rôzne informácie o danom objekte. *Atribúty* môžu mať rôzne hodnoty, ako reťazec, číslo, dátum alebo pravdivostnú hodnotu. Ak by sme si zobrali predchádzajúcu entitu *Bratislava*, jej číselná vlastnosť môže byť napríklad počet obyvateľov.

Vzťahy sú najpodstatnejšou súčasťou ontológie. Poskytujú prepájanie jednotlivých entít. Je to jednosmerné spojenie, ktoré určuje vzťah, v akom sú dve dané triedy. Tým vznikne triplet *trieda:vzťah:trieda*. Medzi triplet sa radí aj trojica *trieda:atribút:hodnota*. Väzby sa zvyknú definovať aj inverzne. Z logického hľadiska sú vzťahy axiómami. Pokiaľ máme triedu *Krajina* a *Hlavné mesto*, tak by vzťah mohol vyzeráť nasledovne: *Krajina:má:Hlavné mesto*.

TBox je množina definícií tried a ich vzťahov medzi nimi. V množine je obsiahnutá znalosť taxonómie tried, taktiež v nej môže byť zadaná disjunktnosť jednotlivých tried, vymenovanie konkrétnych entít obsiahnutých v danej triede, reštrikcie pre jednotlivé triedy a ich vzťahy

ABox je množina znalostí o jednotlivých entitách.

Ontológia má veľa vlastností, ktoré musia byť dodržané. Každý prvok musí byť jasne identifikovateľný. Taktiež zakazuje zapisovanie duplicitných dát, čo nám zaobstará vlastnosť efektívneho ukladania informácií, kde to môže nie len uľahčiť vyhľadávanie ale aj zredukovať obsah pamäti na disku.

3.2 Využitie ontológií

Ontológie sa začali využívať najmä v organizáciách, ktoré sa špecializovali na umelú inteligenciu. Neskôr sa to rozšírilo aj do bežne používaných aplikácií. Napríklad firma Amazon používa ontológie na kategorizovanie tovaru v ich elektronickom obchode.

Ontológie si našli uplatnenie aj v medicínskej oblasti a to napríklad SNO-MED, čo je najväčším viacjazyčným medicínskym slovníkom na svete.

Taktiež sa s ním stretávame každodenne pri vyhľadávaní na stránke Google, kde ako bočný panel sú zobrazené informácie o vyhľadávanom objekte (obrázok nižšie). Tieto dáta je možné zobrazíť preto, lebo výsledkom takéhoto panelu je vyhľadávanie informácií na webovej stránke, ktorá obsahuje sémantické dáta.

Na získavanie dát zo sémantických webov a z RDF úložísk sú využívané SPARQL dopyty. Syntax jazyku SPARQL je veľmi podobná klasickému SQL jazyku, kde aj SPARQL umožňuje okrem dopytovania aj vkladanie, editáciu a vymazávanie dát.

3.3 Syntax ontológií

Popis možnosti

3.3.1 Web Ontology Language

Web Ontology Language alebo OWL

3.4 Deskripčná logika

zakladna syntax ktoru budeme pouzivat neskor na popisanie nasej ontologie

Kapitola 4

Existujúce ontologické riešenia

Množstvo kyber útokov v dnešnej dobe narastá závratnou rýchlosťou, čo značí že dnešné spôsoby a metódy ochrany nie sú dostatočné a preto je potrebné sa zamyslieť nad novými spôsobmi ochrany. Jeden z prístupov by mohol byť založený práve na ontológiách. Ontológie a systémy postavené nad nimi majú výhodu sémantiky, ktorá je schopná rozlišovať situácie kedy je počítačový systém normálny alebo škodlivý.

Problém s ktorým sa ale potýkame je ten, že neexistuje jednotný formát zápisu údajov. Väčšina nástrojov ktoré v dnešnej dobe existujú, majú vlastné štandardy. Keďže tieto štandardy sú prevažne rozdielne, nedá sa ich prepájať a využívať efektívne. Tento problém by mohol byť taktiež vyriešený vďaka ontologickému riešeniu. Tým pádom by sme vedeli mať také dáta, ktoré dokážu stroje nielen prečítať, ale zároveň aj pochopiť.

Ontologický prístup taktiež poskytuje jednoduchšiu rozširiteľnosť už existujúcej ontológie a tým sa dá vytvárať presnejší popis záznamov.

Vďaka URI reprezentáciám jednotlivých entít, ktoré sú používané ako identifikátory jednotlivých objektov, nemôže nastať problém nepochopenia dát ako k tomu môže dochádzať v ľudskej reči. Napríklad ak by sme povedali

slovo *koruna*, nikto nevie, či máme na mysli korunu stromov alebo kráľovskú korunu. Avšak vďaka atribútom vieme toto slovo lepšie pochoiť, keďže nám ho atribúty bližšie definujú.

V súčasnosti existuje veľa rôznych štandardov a ontologických riešení pre doménu kzber bezpečnosti, avšak veľa z nich už ani nie je vyvíjaných. Organizácie, ktoré vyvíjali tieto ontológie, buď stratili o ďalší vývoj záujem, alebo už len nezverejňujú svoje pokroky v danej doméne, teda prešli na closed-source systém.

V nasledujúcich kapitolách si povieme niečo o zaužívaných štandardoch, základnom modeli, z ktorého vychádzame a podľa, ktorého posudzujeme, či je daná ontológia dobrá. Taktiež rozoberieme existujúce riešenia v doméne kyber bezpečnosti.

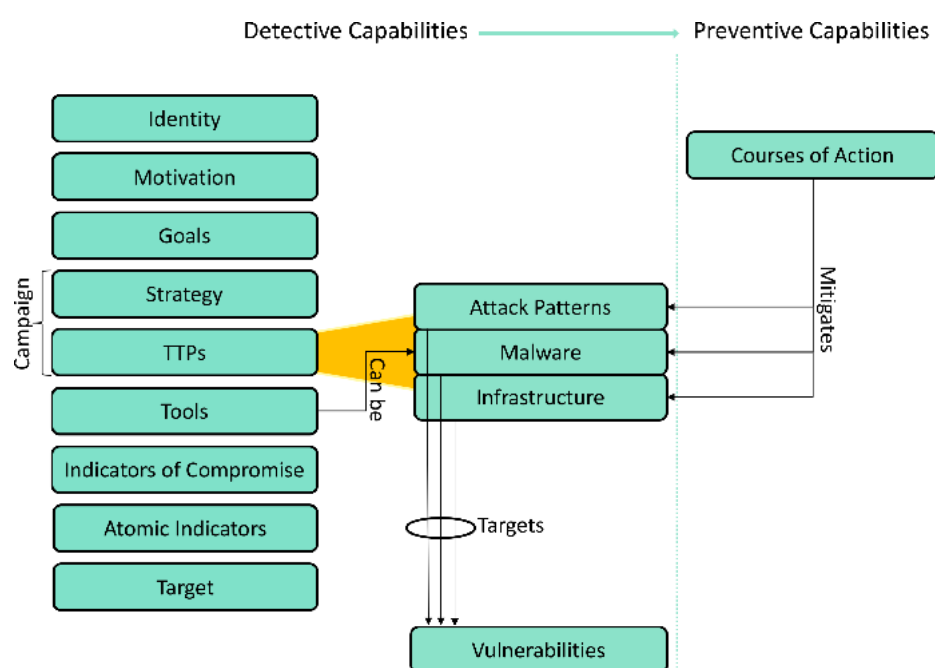
4.1 CTI model

TEN BY SOM MOZNO ZAHRNUL SEM CI? [MB17]

4.2 Integrated Cyber Analysis System

Integrated Cyber Analysis System[SW15] alebo ICAS je ontológia vytvorená pre TAPIO (Targeted Attack Premonition using Integrated Operational data) nástroj, ktorý je schopný extrahovať dáta z počítačov v jednej sieti do jedného sémantického grafu a tým zjednoduší a urýchli prácu bezpečnostným tímom pri vyhľadávaní ohrození systému, čím by sa zvýšila prehľadnosť dát a tiež znížil dopad útoku.

Samotná ontológia ICAS je veľmi komplexnou, nakoľko obsahuje približne 30 podontológií, kde každá sa špecializuje na inú oblasť v doméne



Obr. 4.1: CTI model.

Zdroj: [MB17]

informačnej bezpečnosti.

Nástroj TAPIO spolu s ontológiou ICAS bol vyvíjaný organizáciou DAPRA, ale dátum poslednej úpravy bol v roku 2017, čiže podobne ako UCO sa jedná o projekt ktorý už nie je aktuálny.

4.3 STUCCO

STUCCO [IBN⁺15], ktorej autorom je Iannacone at al. je ontológiou, ktorá je určená na prácu so znalostnými grafovými databázami. Jej základ tvoria scenáre použitia ľudskými používateľmi alebo automatizovanými strojmi. Obsahuje dáta z 13 rôznych štruktúr, ktoré majú rôzne formáty a ktoré sú uložené v rôznych typoch databáz.

STUCCO obsahuje dáta z nasledovných kategórií do ktorých je rozdelená bezpečnostná doména.

- Identita – predstavuje totožnosť a vlastnosti útočníka.
- Taktika technika a procedúry (TPP) – Popisuje čo daný útok robí a ako to robí.
- Nástroje – Aké nástroje sú potrebné pre úspešné vykonanie útoku.
- Atomické indikátory – Sem môžu spadať súbory, IP adresy, doménové mená atď. Nanešťastie tieto dáta majú krátku životnosť nakoľko sa stále menia.

Časť II

Návrh riešenia

Kapitola 5

Unified Cybersecurity Ontology

V tejto časti popíšeme ontológiu, ktorú sme si vybrali pre ďalší vývoj a budeme porovnávať UCO ontológiu vzhľadom na CTI model. Popíšeme taktiež nedostatky danej ontológie vzhľadom na model CTI a naše úpravy v danej ontológii.

5.1 Popis ontológie

Unified Cybersecurity Ontology [SPF⁺16] alebo skrátene UCO je rozšírením pôvodného projektu Intrusion Detection System (IDS), ktorého tvorcom je rovnaká skupina. Spája viaceré bežne dostupné bezpečnostné štandardy používané v kybernetickej bezpečnosti. Prevažne pokrýva STIX, ktorý je najväčším a najkomplexnejším štandardom, pokrývajúcim najväčšiu časť kybernetickej bezpečnostnej domény ale taktiež pokrýva iné relevantné štandardy ako CVE4, CCE5, CVSS6, CAPEC7, CYBOX8, KillChain9 a STUCCO10.

Aj keď je STIX najkomplexnejším štandardom a zjednocuje všetky informácie o kybernetických hrozbách, má tieto dáta uložené v XML súboroch, takže nepodporuje výhody inferencie v ontológiách, čo UCO poskytuje.

Okrem týchto štandardov obsahuje aj mapovanie na všeobecné databázy ako sú Google Knowledge Graph, DBPedia a Yago. Vďaka týmto mapovaniam je možné mať prístup k verejným databázam z rôznych domén záujmu.

Základnými triedami, využívanými v UCO sú:

- *Means* – Čo je zamýšľané daným útokom.
- *Consequences* – Dôsledky útoku.
- *Attack* – Typ útoku.
- *Attacker* – Kto je iniciátorom daného útoku.
- *Attack-Pattern* – Vzorec útoku, podľa ktorého je útok riadený.
- *Exploit* – K čomu útok slúži.
- *Exploit Target* – K čomu slúži cieľ alebo výsledok útoku.
- *Indicators* – Indikátor útoku.

Každá z týchto tried je mapovaná na už reálne existujúcu triedu v niektorom z vyššie uvedených štandardov, prevažne na STIX schému.

Ontológia UCO umožňuje analytikom zachytávať špecifické vedomosti o kybernetickej bezpečnosti pomocou termínov a tried z ontológie a taktiež umožňuje písať pravidlá, ktoré sa môžu použiť na odvodenie nových poznatkov.

Vývojári extrahovali dáta z National Vulnerability Database (NVD), ktorá je uložená v XML súboroch. Potom boli namapované na triple store DBPedia a dáta boli uložené na FUSEKI server, ktorý podporuje dopytovanie z rôznych zdrojov rovnako ako ich odvodzovanie.

5.2 Model

Základnou triedou tejto ontológie je *UCO Thing*, ktorá je nadrtiedou každej triedy v ontológii.

Pokiaľ UCO ontológia využíva už existujúcu triedu z inej ontológie, zdefinuje si ju aj pre svoju doménu a vďaka slovníku OWL jej zdefinuje ekvivalentnú triedu, čím zabezpečí opätovné použitie dát pre iný zdroj. Napríklad trieda *Indicator* je vďaka tomuto vzťahu prepojená s ontológiou CAPEC.

Identita je v ontológii UCO reprezentovaná triedou *Attacker*, ktorá je namapovaná na triedu *ThreatActor* zo STIX-u.

Identita		
Vlastnosť	Doména	Rozsah
hasAssociatedCampaign	Attacker	Campaign
hasConfidenceValue	Attacker	ConfidenceType
hasIntendedEffect	Attacker	StatementType
hasMotivation	Attacker	StatementType
hasObservedMeans	Attacker	Means
hasRelatedIncidents	Attacker	Incident
hasSophistication	Attacker	StatementType
hasType	Attacker	StatementType
hasTitle	Attacker	xsd:string

Útok máme reprezentovaný triedou *Attack*, ktorá má viaceré pravidlá na splnenie. Musí mať minimálne jeden význam (*Means*) a minimálne jeden následok (*Consequence*) útoku. Samotný význam je ekvivaletný s triedou TTP, ktorá popisuje samotný útok pomocou informácií o vzoroch útoku, využívaní slabín systémou alebo známych malwaroch. Nakoľko vďaka jazyku owl vieme definovať napríklad inverzné vlastnosti je žiadúce aby sa táto vlastnosť vy-

užívala. Pre vlastnosť *hasAttacker* to žiaľ nie je zadefinované a určite by to bolo vhodné zadefinovať.

Útok		
Vlastnosť	Doména	Rozsah
hasAttacker	Attack	Attacker
hasConfidenceValue	Attack	ConfidenceType
hasIndicator	Attack	Indicator
hasMeans	Attack	Means
hasObservable	Attack	Observable
hasRequestedCOA	Attack	CourseofAction
hasSource	Attack	Source
hasTakenCOA	Attack	CourseofAction
isLaunchedBy	Attack	Attacker

Literatúra

- [BHBL11] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data: The story so far. In *Semantic services, interoperability and web applications: emerging concepts*, pages 205–227. IGI Global, 2011.
- [BLHL01] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific american*, 284(5):34–43, 2001.
- [GOS09] Nicola Guarino, Daniel Oberle, and Steffen Staab. What is an ontology? In *Handbook on ontologies*, pages 1–17. Springer, 2009.
- [IBN⁺15] Michael Iannacone, Shawn Bohn, Grant Nakamura, John Gerth, Kelly Huffer, Robert Bridges, Erik Ferragut, and John Goodall. Developing an ontology for cyber security knowledge graphs. In *Proceedings of the 10th Annual Cyber and Information Security Research Conference*, pages 1–4, 2015.
- [KM] Marja-Riitta Koivunen and Eric Miller. Semantic web – layers. <https://www.w3.org/2001/12/semweb-fin/w3csw>. Navštívené: 01.02.2020.
- [MB17] Vasileios Mavroeidis and Siri Bromander. Cyber threat intelligence model: An evaluation of taxonomies, sharing standards,

- and ontologies within cyber threat intelligence. In Joel Brynielson, editor, *European Intelligence and Security Informatics Conference, EISIC 2017, Athens, Greece, September 11-13, 2017*, pages 91–98. IEEE Computer Society, 2017.
- [OCM12] Leo Obrst, Penny Chase, and Richard Markeloff. Developing an ontology of the cyber security domain. In *STIDS*, pages 49–56, 2012.
- [OCWM14] Alessandro Oltramari, Lorrie Faith Cranor, Robert J Walls, and Patrick D McDaniel. Building an ontology of cyber security. In *STIDS*, pages 54–61. Citeseer, 2014.
- [PUJF03] John Pinkston, Jeffrey Undercoffer, Anupam Joshi, and Timothy Finin. A target-centric ontology for intrusion detection. In *Procs. of the IJCAI-03 Workshop on Ontologies and Distributed Systems*, 2003.
- [SPF⁺16] Zareen Syed, Ankur Padia, Tim Finin, M. Lisa Mathews, and Anupam Joshi. UCO: A unified cybersecurity ontology. In David R. Martinez, William W. Streilein, Kevin M. Carter, and Arunesh Sinha, editors, *Artificial Intelligence for Cyber Security, Papers from the 2016 AAAI Workshop, Phoenix, Arizona, USA, February 12, 2016*, volume WS-16-03 of *AAAI Workshops*. AAAI Press, 2016.
- [SRa] A.T. Schreiber and Raimond. Ontotext ad, sparql overview. <https://www.w3.org/TR/2013/REC-sparql11-overview-20130321/>. Navštívené: 01.02.2020.

- [SRb] A.T. Schreiber and Raimond. Rdf framework. <https://www.w3.org/TR/rdf11-primer/>. Navštívené: 01.02.2020.
- [SW15] Malek Ben Salem and Chris Wacek. Enabling new technologies for cyber security defense with the icas cyber security ontology. In *STIDS*, pages 42–49, 2015.
- [TPKN18] Takeshi Takahashi, Bhola Panta, Youki Kadobayashi, and Koji Nakao. Web of cybersecurity: Linking, locating, and discovering structured cybersecurity information. *Int. J. Communication Systems*, 31(3), 2018.

Zoznam obrázkov

2.1	Semantic Web - vrstvy. Zdroj: [KM]	4
2.2	Príklad grafovej databázy.	7
4.1	CTI model. Zdroj: [MB17]	16