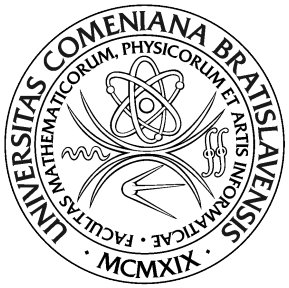
UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

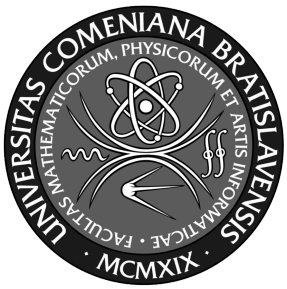



# SÉMANTICKÉ PUBLIKOVANIE SPRAVODAJSKÝCH DÁT O BEZPEČNOSTNÝCH HROZBÁCH

Diplomová práca

2021 Bc. Matej Rychtárik

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY




# SÉMANTICKÉ PUBLIKOVANIE SPRAVODAJSKÝCH DÁT O BEZPEČNOSTNÝCH HROZBÁCH

Diplomová práca

| | |
|---|---|
| Študijný program: | Aplikovaná informatika |
| Študijný odbor: | 2511 Aplikovaná informatika |
| Školiace pracovisko: | Katedra aplikovanej informatiky |
| Školiteľ: | doc. RNDr. Martin Homola, PhD. |
| Konzultant: | Ing. Štefan Balogh, PhD. |

Bratislava, 2021 Bc. Matej Rychtárik


Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

# ZADANIE ZÁVEREČNEJ PRÁCE

**Meno a priezvisko študenta:** Bc. Matej Rychtárik
**Študijný program:** aplikovaná informatika (Jednoodborové štúdium, magisterský II. st., denná forma)
**Študijný odbor:** informatika
**Typ záverečnej práce:** diplomová
**Jazyk záverečnej práce:** slovenský
**Sekundárny jazyk:** anglický

**Názov:** Sémantické publikovanie spravodajských dát o bezpečnostných hrozbách
*Semantic punishing of security threat intelligence data*

**Anotácia:** V súčasnosti zaznamenávame veľké množstvo nových spravodajských dát o rôznych bezpečnostných hrozbách. Pre popis a publikovanie týchto dát vznikli v minulosti viaceré štandardy. Nový trend v oblasti však ukazuje potrebu sémantickej anotácie týchto dát za účelom zvýšenia ich dosahu a interoperability.

**Cieľ:** Cieľom je navrhnúť vhodnú ontológiu pre publikovanie spravodajských dát o bezpečnostných hrozbách a vytvorenie repozitára za týmto účelom v sieti prepojených dát.

**Literatúra:**
[1] Allemang, D. and Hendler, J., 2011. Semantic web for the working ontologist: effective modeling in RDFS and OWL. Elsevier.
[2] Heath, T. and Bizer, C., 2011. Linked data: Evolving the web into a global data space. Synthesis lectures on the semantic web: theory and technology, 1(1), pp.1-136.
[3] Mavroeidis, V. and Bromander, S., 2017. Cyber threat intelligence model: an evaluation of taxonomies, sharing standards, and ontologies within cyber threat intelligence. In EISIC 2017 (pp. 91-98). IEEE.

**Vedúci:** doc. RNDr. Martin Homola, PhD.
**Rektorát, dekanát:** FMFI.Dek - Dekanát

**Dátum zadania:** 02.10.2019

**Dátum schválenia:** 14.10.2019 prof. RNDr. Roman Ďurikovič, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

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

# Poďakovanie

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

Abstrakt v slovenčine

# Abstract

An abstract in english language

# Obsah

# Kapitola 1

# Úvod

Nejaky strucny uvod do problematiky

# Časť I

# Prehľad problematiky

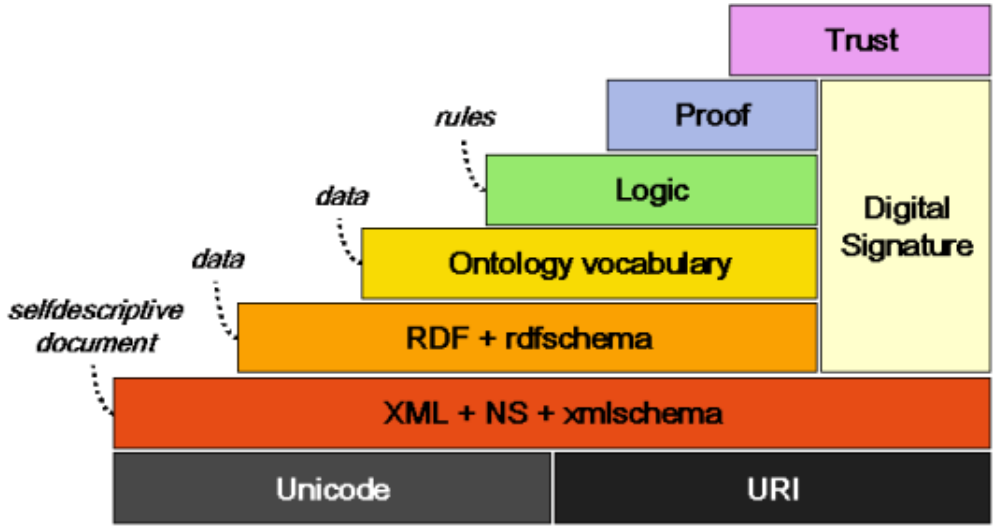
# Kapitola 2

# 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 neštandartné 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 zadefinovaním spojenia medzi objektami ”Jožko” a ”jablko” pomocou




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

vzťahu "má". Toto spojenie je známe ako triplet, ktorý je základnou stavebnou jednotkou sémantického 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 zadefinovať vzťah podtriedy vlastnosťou "rdfs:subClassOff". 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 spĺnať. Vďaka logickej vrstve vieme napríklad zadefinovať 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.

MH: ↑Toto trochu povrchne: (1) Ucelom SW nie je vyssia pouzitelnost webu (to je nepresne), ale je to lepsia pristupnost informacii publikovanych na webe pre strojove spracovanie. (2) Ak chces popisovat vrstvy SW podla tohto diagramu, bolo by dobre keby si popisal vsetky vrstvy – U XML by som sa obmedzil na to, ze je to proste dobry format pre textovu reprezentaciu dat v suboroch a pre ich vymenu medzi softvermi – toto vsak uz je dnes prekonane, uz vymiename SW data aj ako JSON, embedujeme ich do HTML5 (chcelo by to poznamku) – O RDF a RDFS si vlastne nic uzitocne (z coho citatel nieco vyrozumie) nepovedal – no a ostatne vrstvy si uplne preskocil

MH: Toto si menil? Zda sa mi, ze trochu asi aj ano, ale o tych dalsich vrstvach si nic nenapisal

MR: Popisane su uz aj dalsie vrstvy, mali by byt uz hotove

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

## 2.1 Linked Data

MH: ↓Tato sekcia je celkom fajn ale chybalo mi trochu premostenie od SW – linked data bola iniciativa, ze ked uz SW formaty mame, podme v nich aj data zverenjnovat

MH: Inak ked uz si sa rozhodol pisat po slovensky, mal by si pouzivat aj slovensku terminologiu (co je celkom peklo) – ale teda Linked Data su *prepojene data*, LD network je *siet prepojenych dat* – ludia to takto pouzivaju

MR: premenovane

Linked Data [BHBL11] alebo prepojené dáta, je metóda zverejňovania štrukturovaných dát. Ich hlavným cieľom je poprepájať existujúce databázy (primárne písané v RDF formáte), medzi rôznymi údajmi a umožniť ľuďom

zdielať štrukturované dáta na webe pomocou HTML. Časť vízie do budúcna je, aby sa Internet stal globálnou databázou. Princípy prepojených dát 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 Princípy prepojených dát.

MH: Tu mi chyba informacia, ze sa tato inciativa ujala, a ze vdaka tomu vznikla na webe tzv. siet prepojenych dat, ktora obsahuje obrovske mnozstvo datovych zdrojov a nieco viac o tej sieti.

## 2.2 Resource Description Framework (RDF)

RDF [SRb] je štandartný 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 softvérmi, bez straty ich hodnoty. Dáta sa uchovávajú v Triple Store

databázach, ktorých formát je striktne daný. Výhodou je, že dáta môžu byť spracované aj softvérmi, pre ktoré dané dáta neboli vytvorené.

MH: ↑Na co RDF sluzi sa uz citatel dozvedel v skorsich castiach (ked to tam lepsie ozrejmis). Niektore veci, ktore tu ↑pises su nepresne (napr. to o tych metadatach a "dalsich informaciac" alebo o zdrojoch. Tiez o Triple Stores predbiehas, budes o tom pisat neskor... Asi by som to tu skratil a len by som nadviazal, ze RDF je zakladny datovy format pre SW a tu ho popiseme...

MH: ↓Zvysok ide dobrym smerom, je to presne to, co by som si predstavoval, ze tu budes pisat, len by som to chcel vidiet mozno trochu pomenej, podrobnejsie, systmatickejsie prebrate... Na vacsom priestore, mozno postupne ten priklad budovat... Vysvetlit na nom vsetky zakladne moznosti RDF

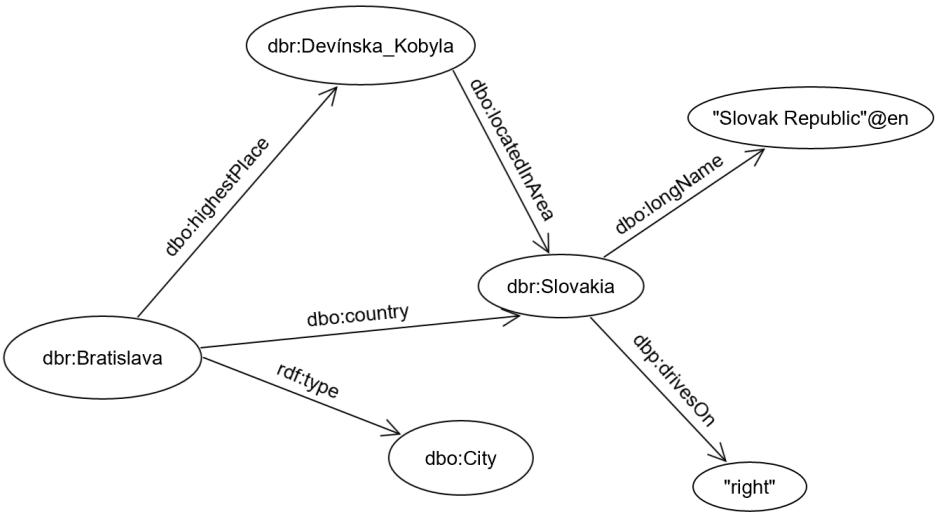
RDF súbor je taký dokument, ktorý ukladá RDF grafy do špecifického formátu serializácie pre RDF, ako sú napríklad N-Triple, TURTLE, RDF/XML a mnohé ďalšie. RDF bol postavený na myšlienke vytvárať údaje vo forme predmet-predikát-objekt, ktorý sa volá triplet. Triplet je základná stavebná jednotka akejkoľvek 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 použítý 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 ontolológií (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 datábazy, 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 vlatnosti výsledných hodnôt dopytu. Ako posledný typ dopytu je CONSTRUCT, ktorý vracia nový RDF graf podľa predlohy vytvorenej v hlave dopytu.

# Kapitola 3

# Ontológie

MH: Predpokladam, ze o ontologiach budeme pisat o znacny kus viac a podrobnejsie... Mozes vychadzat z mojej prednasky ale tiez napr z uvodnej kapitoly *Handbook on Ontologies*

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 nadtriedy a môže byť rozšírená o vlastné vlastnosti.

**Entity** sú individuálne inštancie nejakej nami zadefinovanej 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 vyzerať 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 zadefinovaná disjunktnosť jednotlivých tried, vymenovanie konkrétnych entít obsiahnutých v danej triede, reštrickie 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 indetifikovateľný. 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.

MH: ↑ Neviem, ci zrovna tieto vlastnosi ontologii su tie najpodstatnjsie, a teda treba ich spominat na tomto mieste.
MR: Urcite by som ich spomenul nakolko existuje urcite vela duplicitnych dat a je to vyhoda oproti SQL kde kazdy zaznam v ciselniku ma unikatne IDcko ale u nas je to predstavovane iba entitou.

## 3.2 Využitie ontológií

MH: ↓ Asi by som sa neodvazoval tvrdit to co pises v prvej vete. Ak zoberieme ako ranne vyuzitie napr. SNOMED, nie je to pravda, kedze sa vyuzival v medicinskej praxi. S druhou vetou mozno nesuhlasit. Kto je bezny pouzivatel? Nikomu takemu sa nerozsirila do pocitaca nejaka ontologia (ze by o tom vedel)...
MR: dodefinoval som to ako bezne pouzivane aplikacie.

Ontológie sa začali využívať najmä v organizáciách, ktoré sa šoecializovali 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 SNOMED, čo je najväčším viacjazyčným medicínskym slovníkom na svete.

MH: ↑ Tu chces asi pisat o vyuziti ontologii, takze zmienka o jazyku RDF sem nepatri, navyse RDF nie je jazyk na zapis ontologii (tym je RDFS) a o RDF su uz pisal vyssie

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 zobraziť preto, lebo výsledkom takéhoto panelu je vyhľadávanie informácií na webovej stránke, ktorá obsahuje sémantické dáta.

MH: ↑ SNOMED a Google spominas dobre, a v druhom pripade si nespomenuk ziadnu ontologiu a pritom ju dobre pozname (Schema.org)

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

MH: ↑ toto mi sem opat nesedi, zrejme tomu chces venovat nejaku kratku podsekciu skor v predchadzajucej kapitole (?)
MR: ja som ju tam mal len predtym si mi pisal ze si si este nie isty ci nas to bude zaujimat, tak som to vratil do predchadzajucej casti

MH: ↑ Tuto cast mozno lespie najko pouzit v uvode predch. casti (?)
MR: presunute

## 3.3 Syntax ontológií

Popis moznosti

### 3.3.1 Web Ontology Language

Jazyk Web Ontology Language alebo OWL slúži na vytváranie a definovanie inštancií webových ontológií. Poskytuje nástroje na popis tried,

vlastností a ich samotných inštancií. Oproti klasickým jazykom poskytje možnosť, špecifikovať logické vlastnosti jednotlivých objektov vyskytujúcich sa v ontológií, to znamená, že dokáže popísať aj fakty, ktoré v ontológií nie sú definované priamo, ale sú spojené logickými vlastnosťami. Napríklad, ak máme osoby Janka, Martin a Jozef, kde Janka je rodičom Martina a Jozef je rodičom Janky, pomocou OWL vieme definovať že Jozef je starým rodičom Martina, pričom nepotrebujeme vlastnosť popisujúcu tento vzťah priamo.

### 3.3.2 Druhy OWL

Jazyk OWL poskztuje 3 podjazyky, ktoré sú použitľné v rôznych situáciách.

OWL Lite poskytuje definície na tvorbu hierarchického rozdelenia objektov a jednoduché obmedzenia. Jednoduchými obmedzeniami sa myslí napríklad obmedzenie kardinálnej vlastnosti a to iba na hodnoty 0 a 1.

OWL DL

OWL Full

Každý, kto sa rozhodne vyvýjať ontológiu použitím OWL, by mal zvážiť, ktorý typ je najlepší pre danú ontológiu. Najmä treba zvážiť aké obmedzenia je potrebné pre ich ontológiu definovať.

### 3.3.3 Syntax

Normovaná syntax OWL jazyka je RDF/XML. OWL je jazyk, ktorý rozširuje jazyk RDF.

Základná časť každej ontológie by mala obsahovať menný priestor, kde si definujeme množinu slovníkov používaných v našej ontológií.

```
<rdf: RDF
```

```
  xmlns     = "http://www.semanticweb.org/example"
  xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs= "http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl = "http://www.w3.org/2002/07/owl#"
>
```

Prvá deklarácia definuje menný priestor našej ontológie. Zvyšné deklarácie definujú ďalšie slovníky použité v ontológii.

Po vytvorení menného priestoru si potrebujeme všeobecne definovať našu vlastnú ontológiu. V tejto definícii sa nachádzajú informácie ako komentár k ontológii a názov ontológii a taktiež vieme definovať rôzne ontológie, ktoré budú súčasťou ontológie.

```
<owl:Ontology rdf:about = "">
  <rdfs:comment>Príklad ontológie OWL o osobách</rdfs:comment>
  <owl:priorVersion
     rdf:resource = "http://www.semanticweb.org/example/1.15" />
  <owl:import rdf:resource = "http://www.semanticweb.org/food" />
  <rdfs:label>Ontológia osôb</rdfs:label>
<owl:Ontology>
```

Vlastnosť *owl:import* pridá externú ontológiu do našej pomocou definovaného zdroja cez vlastnosť *rdf:resource*. Pomocou owl:priorVersion vieme definovať verziovanie ontológií.

Ďalej si potrebujeme vedieť definovať triedy v ontológii. Na to nám slúži element owl:Class, v ktorom vieme zadeklarovať definíciu danej triedy pomocou vlastnosti rdf:ID.

```
<owl:Class rdf:ID = "Muž" />
<owl:Class rdf:ID = "Žena" />
```

Hierachické usporiadanie tried vieme ďalej definovať pomocou vlastnosti rdfs:subClassOf, ktorej definujeme jej nadtriedu pomocou vlastnosti rdf:resource. Týchto nadtried vieme definovať viac.

```
<owl:Class rdf:ID = "Manžel">
  <rdfs:subClassOf rdf:resource = "Muž">
  ...
</owl:Class>
```

Pre triedu vieme definovať aj slovný popis a názov. Tento názov vieme definovať v rôznych jazykoch použitím vlastnosti xml:lang. Vlastnosť rdfs:label definuje pre bežného človeka ľahko čitateľný názov triedy. Najčastejšie sa využíva táto vlastnosť pri prezentačných nástrojoch ontológií.

```
<owl:Class rdf:ID = "Manžel">
  <rdfs:subClassOf rdf:resource = "Muž">
  <rdfs:label xml:lang="en">Husband</rdfs:label>
  <rdfs:label xml:lang="sk">Manžel</rdfs:label>
  ...
</owl:Class>
```

MH: Kedze cela Tvoja praca je o ontologiach a budes ich zrejme nejako zapisovat mozno by si mohol nieco povedat aj o jazykoch na zaspis ontologii, minimalen apson o jednom, s ktorym budes pracovat dalej (cize zrejme OWL)

# 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. Tneto 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.

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

Vďaka URI reprezentácií jednotlivých entít, ktoré sú používané ako identifikátory jednotlivých objektov, nemôže nastať problém nepochopenia dát ako k tomu môže doochádzať 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.

MH: ↑ Tento text sa skor hodi do uvodu, pripadne cast do casti kde opisujes ontologie. Tuto kapitolu by som ocakaval, ze otvoris nejak zhruba odtialto ↓ (Akoze toto ↑ sa uz citatel (mal) dovediet niekde predtym a je zbytocne to tu opakovat...)

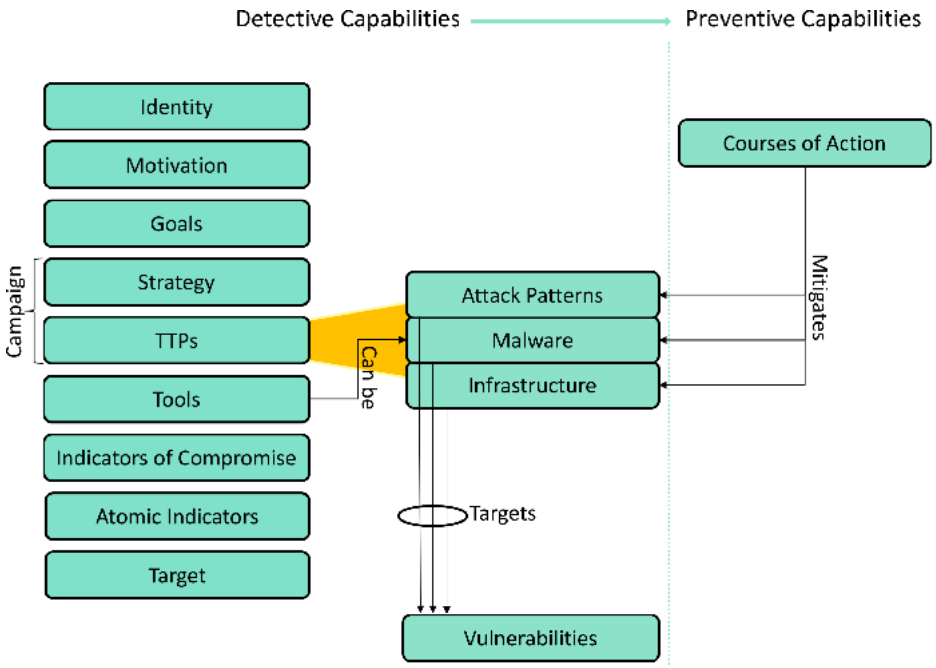
V súčastosti existuje veľa rôznych štandardov a ontologických riešení pre doménu kzber bezpečnosti, avšak veľa z nich už ani nie je vyvýjaných. Organizácie, ktoré vyvýjali tieto ontológie, buď stratili o ďalší vývoj záujem, alebo už len nezverejňujú svoje pokroky v danej doméne, teda prešli na closed-source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

Cyber Threat Intelligance model [MB17] (CTI model) má za cieľ objasňovať rôzne typy informácií, ktoré potrebujú organizácie zhromažďovať o kybernetických hrozbách. Cieľom modelu je jednotne definovať a podrobne popísať okruhy v danej doméne. Výhodou takéhoto modelu je zlepšenie schopností pri prevencii, detekcii a reakcii na bezpečnostné hrozby.

Tento model bol vyvinutý so zámerom, aby slúžil ako predloha pre ontologické riešenia, nakoľko autori zahrnuli fakt, že ontologické riešenia sú najlepšie pri publikovaní a zdieľaní dát a sú strojovo čitateľné vďaka typu reprezentácie.




Obr. 4.1: CTI model.
Zdroj: [MB17]

Model zahŕňa informácie o autorovi útoku, motivácie a ciele útoku, stratégiu akou je útok vedený, aké technológie a postupy používa, ktorým základným vzorcom útoku sa riadi, aké slabiny jednotlivé útoky využili, aké atomické indikátory dokážu predpovedať útok a akú cieľovú skupinu útok postihuje.

V nasledujúcich častiach popíšeme niektoré okruhy z obr. 4.1, ktorý znázorňuje jednotlivé prvky modelu.

### 4.1.1 Identita

Identita môže byť definovaná rôznymi typmi identifikátorov. Ak sa pozná presná totožnosť osoby alebo organizácie, ktorá útok iniciuje, identifikátorom je názov osoby alebo spoločnosti.

Pokiaľ takýto identifikátor nevieme určiť, prideľujeme im anonymnú identitu, ktorú následne vieme spájať vyhodnotením útoku. Pokiaľ zoberieme do úvahy všetky ostatné časti CTI modelu a vyhodnotíme zhodujúce sa časti, vieme povedať, za ktoré útoky zodpovedá rovnaká osoba alebo organizácia.

### 4.1.2 Kampaň

Kampaň popisuje samtotný útok z dvoch pohľadov. Netechnický, ktorý je reprezentovaný stratégiou a technický, ktorý definuje taktiky, techniky a procedúry využívané v útoku.

#### Stratégia

Stratégia predstavuje netechnický popis útoku na hornej úrovni. Pokiaľ si zoberieme nejakého útočníka, tak tento útočník zvyčajne dosahuje ciele útoku viacerými spôsobmi a stratégia definuje, čo a ako je potrebné pozorovať.

### 4.1.3 Zraniteľnosť

Zraniteľnosti definujú slabiny systému. Tieto slabiny môžu byť definované rôznymi stavmi zariadenia ako napríklad, ktoré systémy bežia, ale môžu predstavovať aj nastavenia rôznych aplikácií, napríklad rôzne prístupové nastavenia a povolenia pre webové prehliadače. Taktiež môžu popisovať rôzne funkcie, ktoré umožňujú vykonávať jednotlivé systémy, ako napríklad spúšťanie rôznych príkazov ktoré ovplyvňujú a menia systémové nastavenia zariadenie.

**TTP**

Taktiky, techniky a procedúry (TTP) sa zamieravajú na dáta čitateľné strojmi. Popisujú spôsoby útoku z hľadiska toho čo chcú dosiahnuť a ako to robia.

**Vzorec útoku** je jedným z hľadisiek TTP. Definuje akým spôsobom je útok použitý. Napríklad sa tu nachádzajú rôzne úpravy názvov súborov, rôzne prenastavenie časovania a stavov zariadenia, atď.

**Malware** je taktiež hľadisko z TTP. Definuje typ softvéru, ktorý sa vkladá do systému s úmyslom poškodiť cieľ útoku z hľadiska dôvernosti, integrity alebo dostupnosti. Medzi typické malware patrí vírus, trójsky kôň, červ, metódy tajných vstupov, spyware atď.

Každý okruh z TTP využíva niektoré zraniteľnosti systémov.

### 4.1.4 Indikátor

Indikátory predstavujú množinu rôznych vlastností zariadenia, kedy je zariadenie náchylné na útok. Tieto indikátory popisujú rôzne nastavenia zariadenia, jeho technické parametre alebo nainštalovaný softvér a vďaka nemu sú schopné definovať rôzne útoky.

**Atomické indikátory** sú najpremenlivejšie ukazovatele, nakoľko sa po čase menia. Napríklad ak niektoré útoky prebiehajú z určitej IP adresy, tak táto adresa sa môže časom zmeniť. Tieto typy dát môžu pomáhať pri identifikovaní útočníka iba v určitom čase.

### 4.1.5 Nástroj

Nástroje sú pomôcky, ktoré útočníci inštalujú v zariadení obete. Väčšinou zahŕňa špecializovaný softvér priamo určený na spôsobenie škôd, ako

napríklad vzdialené vykonávanie procesov alebo sieťové skenovanie. Taktiež môže byť použitý pre zabránenie detekcie útoku inými systémami.

### 4.1.6 Zámer

Zámer pomenováva koncový stav objektu útoku. Zámer nemusí byť vopred daný alebo zrejmý. Jednotlivé útoky môžu prebiehať aj s tým, že až počas útoku je zámer známy. Zámerom môže byť napríklad ukradnutie duševného vlastníctva, poškodenie systému, získanie kompromitujúcich dát a podobne.

### 4.1.7 Cieľ

Cieľom je myslená entita, na ktorú je útok namierený. Takouto entitou môže byť napríklad organizácia, štát alebo konkrétna osoba.

MH: Ano, urcite

MR: zahrnute, plus som sem vypichol casti ktore budeme popisovat z modelu a na ktore budem vyhodnococat UCO podrobne

## 4.2 Unified Cybersecurity Ontology

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 relevatné 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 pomocov 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.

MH: ↑ Zmienky o nejakych datach sa tu zjavia z cista-jasna... Doteraz sme hvorili stale o ontologii, nezapada to... (Mozno treba len previazat a upresnit?)

## 4.3 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 informačnej bezpečnosti.

MH: ↑ vieme povedat, ze ci nejaka cast z toho je relevantna pre nas

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

MH: ↑ Tu by som bol opatrnejsi, tvrdit, ze nieco nie je aktualne, lebo posledny update bol pred 3 rokmi je zvlastne. Co ked pred troma rokmi to dotiahli uz do dokonalosti a teraz uz len pouzivaju?

## 4.4 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 vykonaie útoku.
- Atomické indikátory – Sem môžu spadať súbory, IP adresy, doménové mená atď. Nanešťastie tieto dáta majú krátku životnosť nakoľko sa stále menia.

# Časť II

# Vlastný prínos

# Kapitola 5

# Výskum a Analýza UCO

MH: Takze si to vlastne sem cele presnul a v Casti I uz o tom nie je ani zmienka? Tu by to malo byt uvedene sposobom, ze tu ju podrobne pispiseme a zanalyzujeme... Cize malo by to byt v Casti II vedene ako analyza UCO, nie jej len popis
MR: Done
MH: Namiesto ↓ „popiseme a budeme porovnavat“ napise „podrobne analyzujeme a vyhodnocujeme vzhladom na“.. Nech je to este jasnejsie.
MR: Done

V tejto časti podrobne analyzujeme a vyhodnocujeme ontológiu[Lab17], ktorú sme si vybrali pre ďalší vývoj a budeme vyhodnocovať UCO ontológiu vzhľadom na CTI model. Pre túto prácu sú pre nás najpodstatnejšie časti Identita, Útok, Kampaň (Stratégia a TTP), Slabina, Indikátor a Nástroj. Na tieto oblasti sa zameriame v nasledujúcich kapitolách, vyhodnotíme, či sú v ontológii popísané dostatočne, či sú prepájané s už existujúcimi ontológiami a či sú ich vlastnosti a hierarchia dostatočné.

## 5.1 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, zadefinuje si ju aj pre svoju doménu a pomocou jayzka OWL jej zadefinuje ekvivalentnú triedu, čím zabezpčí 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*, ďalej útočník, ktorý je namapovaný na existujúcu triedu *ThreatActor* zo STIX-u.

MH: Mozno by sme mohli na obrazku tie triedy z inych ontologii nejako lepsie vyznacit? Je blbost vyrobit vacsi obdlznikovity biely tvar s nadpisom STIX zapuzdujuci veci zo STIXU (hoc by tam bola aj len 1 trieda)?

Každý útočník má nejaké meno (hasTitle) a je priradený k existujúcim incidentom reprezentovaných triedou *Incident* priradených vlastnoťou *hasRelatedIncident*. Má aj určitú mieru doveryhodnosti (hasConfidenceType). Každý útočník má priradené nejaké zavedené postupy a stratégie pri útoku, ktoré sú inštanciou triedy *Campaign* priradené vlastnosťou *hasAssociatedCampaign*.

MH: ↕ chyba premostenie na tabulku vlasntosti, mozno aj viac ich popisat v texte (?)

**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 systémov alebo známych malwaroch. Nakoľko vďaka jazyku OWL vieme definovať napríklad inverzné vlastnosti je žiadúce aby sa táto vlastnosť využívala. Pre vlastnoť *hasAttacker* to žiaľ nie je zadefinované a

| Identita – Attacker | | |
|---|---|---|
| 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 |

Tabuľka 5.1: Tabuľka vlastností popisujúcej časť **Identita** z CTI modelu.

určite by to bolo vhodné zadefinovať.

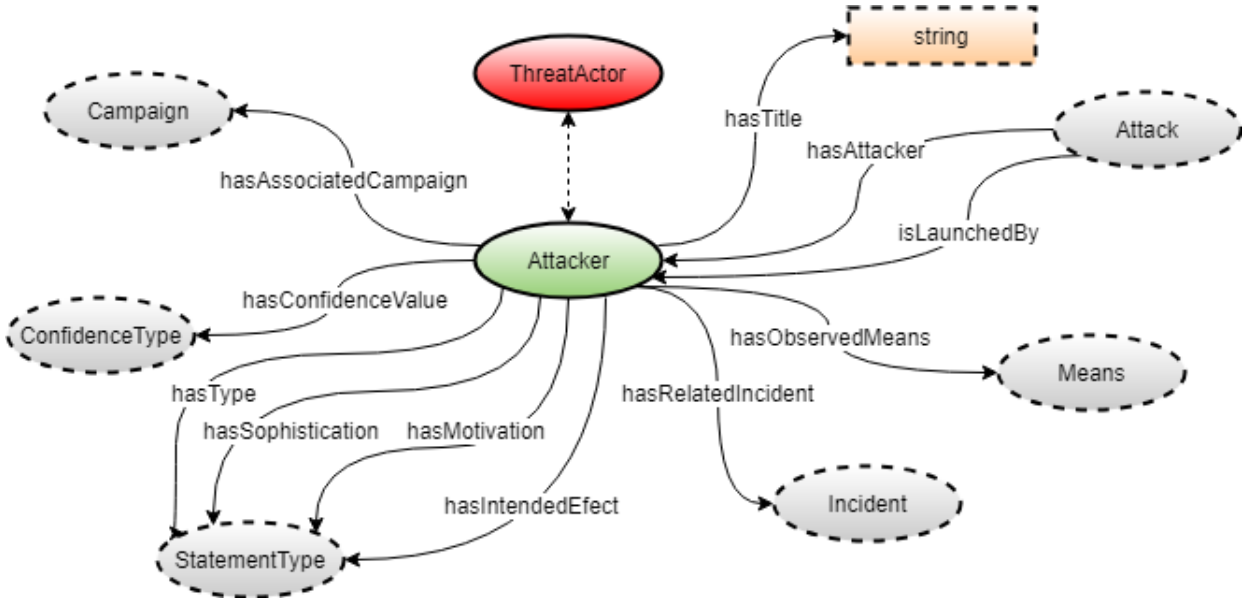
MH: ↑ Tu zasa strasne skaces, viac vysvetluj... Kde sa vzalo TTP? Tiez pises, ze *hasAttacker* nie je definovane, ale v tabulke to je...

MR: tym som povedal ze nie je zadefinovana ziadna inverzna vlastnost k vlastnosti *hasAttacker*

MH: Ano, ale je to tu take vlepene, citatel nechape, preco prave tu uvadzas takuto poznmaku, navyse, asi sa to tyka aj inych vlasnosti, ku ktorym by sa dala dodefinovat inverzna (?) A dalsia vec, nadvazujeme niekde na to? Navrhujeme vylepsenia tohto druhu?

| Útok – Attack | | |
|---|---|---|
| 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 |

Tabuľka 5.2: Tabuľka vlastností popisujúcej časť **Útok** z CTI modelu.




Obr. 5.1: UCO ontológia – časť Identita

**Kampaň** je zahrnutá v triede *Campaign*, ktorá môže existovať iba v prípade, že už bola použitá v nejakom útoku. Každá kampaň môže taktiež mať nejakú ďalšiu kampaň, ktoré medzi sebou súvisia, kde táto vlastnosť je obojsmerná. Taktiež má vlastnosť hasCampaign, ktorej doménou je trieda *Indicator*. Táto trieda pochádza z ontológie CAPEC, ktorá obsahuje známe vzory útokov.

MH: ↕ Tu (ale aj inde) je taky problem, ze v tabulke je mnozstvo vlastnosti triedy Attack, ci Campaign, ale v texte vysvetlujes len niektore. Nemusis vsetky vymenovavat, ale mozno by sa dalo aspon zhrnut, zgurpit ich... Nieco ako: Ostatne vlasntosti triedy Attak definuju/umoznuju priradit/... ...

**Slabiny** sú zahrnuté v triede *Vulnerability*, ktorá nie je namapovaná na žiadnnu existujúcu ontológiu, avšak dáta už existujú v rámci databázy CVE, ktorá predstavuje dáta o softvérových a hardvérvých chybách. Tieto slabiny sú naviazané na objekty typu *Product* vlastnosťou *affectsProduct*, ktoré môžu byť buď hardvérové alebo softvérové a každá slabina ovplyvňuje


Obr. 5.2: UCO ontológia – časť Útok

| Kampaň – Campaign | | |
|---|---|---|
| Vlastnosť | Doména | Rozsah |
| hasAssociatedCampaign | Campaign | Campaign |
| hasCampaign | Indicator | Campaign |
| hasIndicator | Campaign | Indicator |
| hasIncident | Campaign | Incident |
| hasMenas | Campaign | Means |
| hasStatus | Campaign | |
| isLaunchedBy | Campaign | Attacker |
| usesAttacks | Campaign | Attack |

Tabuľka 5.3: Tabuľka vlastností popisujúcej časť **Kampaň** z CTI modelu.

niektorý produkt. Taktiež je v nej zaznamený čas objavenia (*discoveryTime*), spôsob narušenia alebo preniknutia do systému (*hasAccessVector*), zložitosť preniknutia do systému (*hasAccessComplexity*), rôzne typy dopadov alebo skóre úspešnosti (*score*). Môže niesť aj informáciu o zdroji slabiny. Taktiež definuje objekty, ktoré je potrebné pozorovať, ktoré sú typu *Observable*. Táto trieda zatiaľ nie je namapovaná na žiadnu existujúcu, ale vieme že existuje ontológia CyBox, ktorá tieto dáta uchováva.

| Slabina – Vulnerability | | |
|---|---|---|
| Vlastnosť | Doména | Rozsah |
| affectsProduct | Vulnerability | Product |
| discoveryTime | Vulnerability | xsd:dateTime |
| exploitsVulnerability | Means | Vulnerability |
| hasAccessComplexity | Vulnerability | xsd:string |
| hasAccessVector | Vulnerability | xsd:string |
| hasConsequences | Vulnerability | Consequence |
| hasCVE_ID | Vulnerability | CVE |
| hasMeans | Vulnerability | Means |
| hasObservable | Vulnerability | Observable |
| publishedDateTime | Vulnerability | xsd:dateTime |
| score | Vulnerability | xsd:float |

Tabuľka 5.4: Tabuľka vlastností popisujúcej časť **Slabina** z CTI modelu.


Obr. 5.3: UCO ontológia – časť Slabiny

MH: CVE -> CWE?

Taktiež existuje trieda *CWE*, ktorá je prepojená s triedou *Weakness* z už existujúcej CWE databázy. Taktiež ako trieda *Vulnerability*, popisuje slabiny z CTI modelu. Trieda *CWE* popisuje známe typy slabostí hardvérov a softvérov. Obsahuje informáciu o čase zistenia (*timeOfIntroduction*) a stručný popis (*description*). Zvyšné vlastnosti nemajú definované rozsahy, čo je výrazným nedostatkom a je potrebné tieto vlastnosti dodefinovať.

MH: ↑ V tabulke 5.4 vsak ale vsetky vlastnosti maju aj rozsah, o ktore zvysne teda ide?

| Slabina – CWE | | |
|---|---|---|
| Vlastnosť | Doména | Rozsah |
| timeOfIntroduction | CWE | xsd:dateTime |
| discoveryTime | CWE | xsd:dateTime |
| commonConsequences | CWE | Consequences |

Tabuľka 5.5: Tabuľka vlastností popisujúcej časť **Slabina** z CTI modelu.

**Indikátory** sú reprezentované triedou *Indicator*. Táto trieda je namapovaná na už existujúcu triedu z ontológie CAPEC. Indikátory spadajú pod kampane (*hasCampaign*) a majú určitý dopad (*hasImpact*). Každý indikátor má aj význam (*hasMeans*) a objekty na pozorovanie z triedy *Observable* (*hasObservable*). Každý indikátor môže mať nejaký iný indikátor, ktorý s ním súvisí (*hasRelatedIndicator*). Obsahuje aj navrhovaný postup reprezentovaný triedou *CourseOfAction* (*hasSuggestedCOA*).

**Nástroje** TODO – chcem sa pobavit s panom baloghon co by radil medzi ne z uco ontologie, nakolko sa nevidim tolko do tejto domeny aby som vedel povedat ktora do toho spada.

| Indikátory – Indicator | | |
|---|---|---|
| Vlastnosť | Doména | Rozsah |
| hasCampaign | Indicator | Campaign |
| hasConfidenceValue | Indicator | ConfidenceType |
| hasImpact | Indicator | StatementType |
| hasIndicator | CWE | Indicator |
| hasKillChainPhase | Indicator | KillChainPhase |
| hasMeans | Indicator | Means |

Tabuľka 5.6: Tabuľka vlastností popisujúcej časť **Indikátor** z CTI modelu.

# Kapitola 6

# Návrh ontológií

V predchádzajúcej časti sme si podrobnejšie predstavili vybranú časť UCO ontológie. Ukázali sme si, aké časti z CTI modelu spĺňa a aké dáta jednotlivé okruhy reprezentujú. Touto podrobnejšou anlýzou sme zistili, ktoré dáta ešte nepokrývajú alebo sa nemapujú na už existujúce dáta.

V tejto kapitole si ukážeme databázy CVE a OVAL, ktoré predstavujú z CTI modelu časť Zraniteľnosť. Predstavíme si nami vytvorené ontológie, ich mapovania a vlastnosti jednotlivých objektov.

## 6.1 CVE

Common Vulnerabilities and Exposures(CVE) je databáza všetkých chýb zabezpečenia v oblasti kybernetickej bezpečnosti. Jej úlohou je identifikovať, definovať a katagolizovať tieto chyby, ďalej zraniteľnosti. Každá zraniteľnosť má vlastný záznam, ktorý prechádza fázami v databáze. Zraniteľnosť je najprv objavená, potom pridelená a nakoniec zverejnená. Zverejňovať tieto zraniteľnosti môžu iba tí, ktorí majú partnerstvo s programom CVE. Tieto záznamy sú následne používané pri komunikácii a

riešení rôznych bezpečnostných hrozieb, aby sa zabezpečilo rovnaké chápanie problému, ktoré bolo konzistentne definované v CVE.

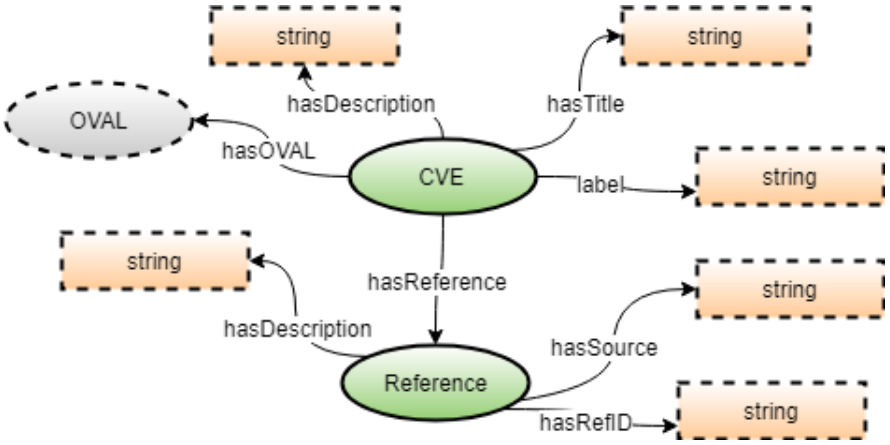
Tieto záznamy majú svoje identifikátory, ktoré sa skladajú z prefixu *CVE*, následne obsahujú rok objavenia a nakoniec unikátny identifikátor v rámci tejto trojkombinácie. Napríklad identifikátor *CVE-2020-0035* predstavuje nejakú zraniteľnosť z roku 2020 v databáze CVE.

Každý jeden záznam má taktiež popis, ktorý musí obsahovať dostatok informácií o tom, na ktoré systémy má daná zranitenosť vplyv, pokiaľ sa vzťahuje aj na konkrétnu verziu, tak aj táto verzia by mla byť uvedená. Musí obsahovať aj infomácie o type zraniteľnosti, jej príčinu a dopad. Daný popis musí byť napísaný v angličtine.

Záznamy majú aj zoznam referencií, ktoré sú používané ako externé zdroje k zraniteľnosti. Sú to externé odkazy s infomáciou o zdrojovom systéme. Tieto odkazy musia používať iba protokoly *http*, *ftp*, *https*, alebo *ftps*.

### 6.1.1 Ontológia

Na základe týchto informácií sme vytvorili ontológiu, ktorá obsahuje základné inforácie zraniteľnosti, podľa požiadaviek štandardu. Ako definíciu zraniteľnosti, používame jej URI, ktorá poskytuje prepoužitie a odkazovanie sa na ňu. Táto zraniteľnosť má taktiež svoj názov, ktorý reprezentujeme ako CVE identifikátor. Každá zraniteľnosť má svoj popis a zoznam referencií. Ako rozšírenie sme použili vlastnosť *hasRefId*, ktorá definuje unikátny identifikátor, v referencovanom systéme. Tento parameter nie je povinný.




Obr. 6.1: CVE ontológia

## 6.2 OVAL

Open Vulnerability and Assessment Language(OVAL) je štandard, ktorý umožňuje popísanie zraniteľností jednotlivých produktov (softvérov). Tieto informácie sú písané vo formáte XML. Obsahuje tri hlavné kroky posudzovania.

Prvým krokom sú základné informácie o systémoch, ako napríklad o aký operačný systém sa jedná, ktoré jeho verzie ohrozuje a aké konkrétne produkty sú ohrozené, pokiaľ je splnená konfigurácia rôznych systémov. Taktiež obsahuje rôzne odkazy na ďalšie informácie o jednotlivých zraniteľnostiach.

Druhé posudzovanie analyzuje systém podľa špecifikovaného nastavenia a zisťuje či je daný systém ohrozený alebo nie.

Posledný krok tvorí správu o výsledku tejto analýzy. Na výstupe je informácia o aktuálnom konfiguračnom nastavení systému.

### 6.2.1 Ontológia

Pre našu ontológiu nie je potrebné reprezentovať všetky znalosti o OVAL-e. Vybrali sme si konkrétne časti, ktoré je potrebné publikovať v našich spravodajských dátach.

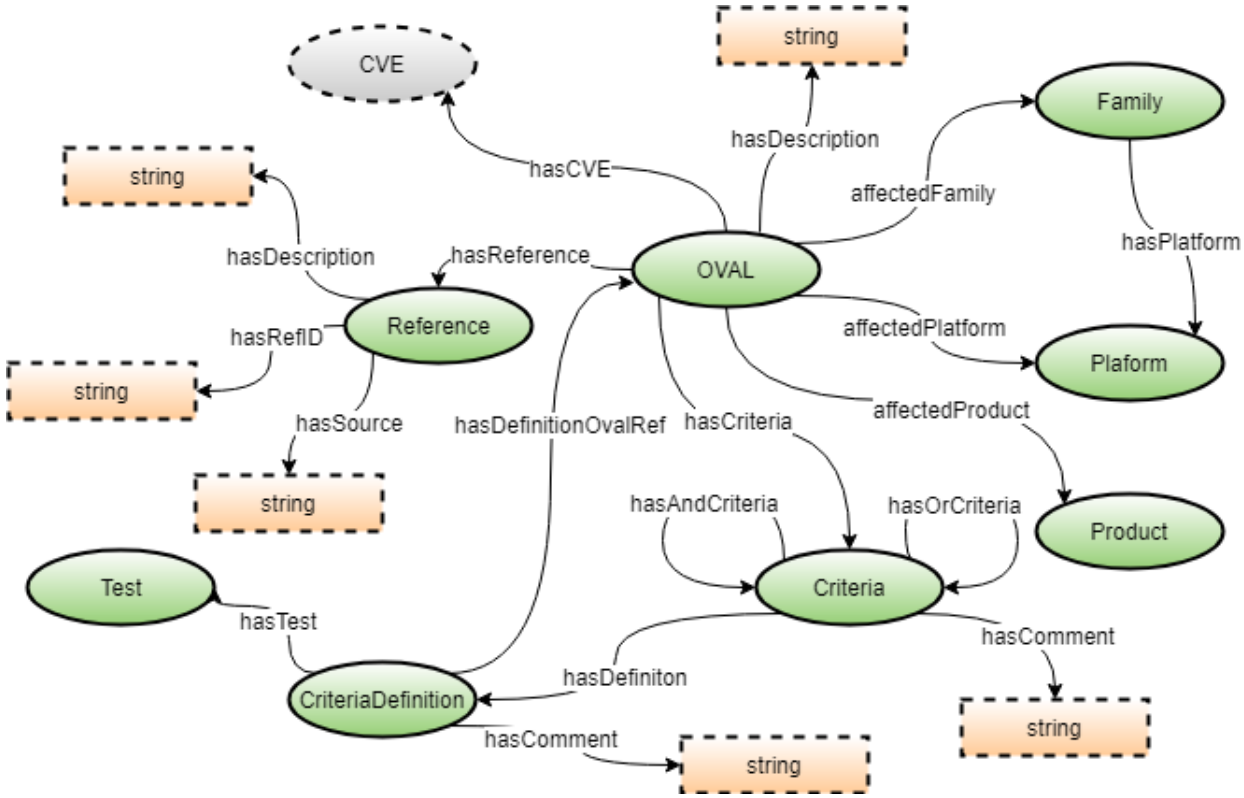
V ontológii reprezentujeme všeobecné informácie popisujúce a definujúce OVAL záznamy, ich vzťahy s jednotlivými operačnými systémami, ich verziami a produktami. Zahŕňame aj informácie o externých zdrojoch pre záznamy. Taktiež predpripravíme ontológiu na reprezentovanie kritérií, ktoré predstavujú logické testy. Tieto testy majú v sebe aj výrokovú logiku reprezentovanú pomocou logických operátorov AND a OR, kde je možné definovať viacúrovňové logické výrazy.

Každý OVAL objekt je reprezentovaný pomocou URN *oval:org.cisecurity:{def}:127* kde *{def}* musí byť z nasledujúcich možností:

- **def** – jedná sa o záznam definujúci zraniteľnosť
- **tst** – záznam, ktorý definuje test

Samotný záznam sa skladá z:

- *hasTitle* – názov záznamu
- *hasDescription* – popis záznamu
- *affectedFamily* – operačný systém, ktorý ovplyvňuje
- *affectedPlatform* – akú platformu operačného systému ovplyvňuje
- *affectedProduct* – ktorý produkt ovplyvňuje
- *hasReference* – odkaz na externý zdroj informácií súvisiaci so záznamom




Obr. 6.2: OVAL ontológia

- *hasCriteria* – zoznam logických testov potrebených na zistenie či daný záznam ovplyvňuje systém

Kritériá popisujúce test majú nasledujúce vlastnosti:

- hasAndCriteria – logické pravidlo spojené s rodičom AND operátorom
- hasOrCriteria – logické pravidlo spojené s rodičom OR operátorom
- hasDefinition – definícia kritéria
- hasComment – popis kritéria alebo definície kritéria
- hasTest – definícia vykonania a kontroly výstupu z testu

# Kapitola 7

# Implementácia a testovanie

V kapitole Návrh ontológií sme si predstavili nami vytvorené ontológie z už existujúcich databáz CVE a OVAL.

V tejto kapitole si ukážeme ako sme na tieto ontológie namapovali už existujúce dáta, ako sme tieto dáta migrovali a ako ich reprezentujeme na webe.

## 7.1 Dáta

Dáta, ktoré sme spracovávali pochádzajú z verejne dostupného zdroja štandardov OVAL a CVE. Tieto dáta mali pôvodne štruktúru XML, ktorá ale nebola písaná OWL syntaxov. Tieto dáta sme spracovali pomocou programovacieho jazyka Python. Jednotlivé triedy v nami navrhnutých ontológiách sme si implementovali a následne sme spracovali dané dátové súbory pomocou knižnice *xml*. Po spasovaní dát sme tieto dáta definovali pomocou tripletov a následne ich zapísali pomocou knižnice *rdflib* do súboru vo formáte *turtle*.

Tento formát sme vybrali najmä kôli menšiemu objemu dát, nakoľko bolo

potrebných viac optimalizácií pre generovanie. Ale aj po tejto optimalizácii sme nevedeli spracovať dané záznamy, preto sme sa rozhodli ich rozdeliť. CVE dáta sme spracovali tak, že sme ich rozdelili podľa rokov vytvorenia. OVAL dáta sme rozdelili podľa operaného systému, ktorého sa týkajú. Po zapísaní časti dát, sme túto časť premazali.

## 7.2 Web

V spolupráci so študentami môjho konzultanta, sme vytvorili webovú stránku, založenú na našej ontológii a našich vygenerovaných dátach. Táto stránka poskytuje zobrazenie CVE a OVAL dát, kde je možné tieto dáta prezerať a vyhľadávať. Pomocou SPARQL dopytov sa vyhľadávajú jednotlivé dáta a ich konkrétne informácie.

## 7.3 Testovanie

V predchádzajúcich častiach sme si popísali ako sme dáta migrovali a mapovali na nami vytvorenú ontológiu. Taktiež sme si popísali webovú stránku, ktorú sme vytvárali v spolupráci so študentami z FEI STU, ktorá slúži na zobrazovanie našich dát.

V tejto kapitole sa budeme venovať testovaniu našej vygenerovanej ontológie a dopytovanie sa nad nami importovanými dátami. Ukážeme si rôzne testy použiteľnosti a ukážky SPARQL dopytov.

### 7.3.1 Testy použiteľnosti

#### Test použiteľnosti 1

**Názov**

CVE podľa platformy

**Zadanie**

Podľa zadanej platformy, nájdi v databáze všetky CVE objekty, ktoré súvisia s danou platformou. Na výstupe vráť ich URI definíciu s názvom CVE záznamu.

**Dopyt**

```
PREFIX main: <http://www.semanticweb.org/rycht/ontologies/cyber_security_ontolog
PREFIX platform: <http://www.semanticweb.org/rycht/ontologies/cyber_security_ont
PREFIX cve: <https://cve.mitre.org/about/terminology.html#>
PREFIX oval:<https://oval.mitre.org/language/version5.11/OVAL>

SELECT ?cve ?title
WHERE {
  ?cve a cve:CVE.
  ?cve main:hasTitle ?title.
  ?oval a oval:.
  ?oval main:hasCVE ?cve.
  ?oval main:affectedPlatform platform:Microsoft_Windows_8
}
```

### Test použiteľnosti 2

#### Názov

OVAL záznamy, ktoré ovplyvňujú iba platformu.

#### Zadanie

Vráť mi všetky OVAL záznamy, ktoré ovplyvňujú všeobecne platformu a žiaden konkrétny produkt pre danú platformu. Na výstupe mi pošli ich URI OVAL záznamu, názov, URI a názov operačnénho systému a URI a názov platformy.

#### Dopyt

```
PREFIX main: <http://www.semanticweb.org/rycht/ontologies/cyber_security_ontolog
PREFIX cve: <https://cve.mitre.org/about/terminology.html#>
PREFIX oval:<https://oval.mitre.org/language/version5.11/OVAL>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?oval ?title ?family ?familyName ?platform ?platformName
WHERE {
  ?oval a oval:.
  ?oval main:hasTitle ?title.
  ?oval main:affectedFamily ?family.
  ?family rdfs:label ?familyName.
  ?oval main:affectedPlatform ?platform.
  ?platform rdfs:label ?platformName.
  NOT EXISTS {
    ?oval main:affectedProduct ?product.
```

```
  }
}
```

### Test použiteľnosti 3

#### Názov

Frekvenčná tabuľka CVE záznamov.

#### Zadanie

Získaj všetky CVE záznamy, zoradené podľa toho, ako často sú využívané ako referencia v OVAL záznamoch. Zoraď ich podľa využívanosti od najväčšej a na výstup vráť URI a názov CVE záznamu s počtom OVAL definícií, v ktorých sú použité.

#### Dopyt

```
PREFIX main: <http://www.semanticweb.org/rycht/ontologies/cyber_security_ontolog
PREFIX platform: <http://www.semanticweb.org/rycht/ontologies/cyber_security_ont
PREFIX cve: <https://cve.mitre.org/about/terminology.html#>
PREFIX oval:<https://oval.mitre.org/language/version5.11/OVAL>

SELECT ?cve ?title (COUNT(?oval) as ?count)
WHERE {
  ?cve a cve:CVE.
  ?cve main:hasTitle ?title.
  ?oval a oval:.
  ?oval main:hasCVE ?cve.
}
GROUP BY ?cve ?title
```

```
ORDER BY DESC(?count)
```

### Test použiteľnosti 4

#### Názov

Cve záznamy ovplyvňujúce viac operačných systémov.

#### Zadanie

Nájdi také CVE záznamy, ktoré ovplyvňujú rovnaký produkt na rôznych operačných systémoch. Na výstup vráť URI a názov CVE záznamu, produkt ktorý ovplyvňuje a URI operačných systémov.

#### Dopyt

```
PREFIX main: <http://www.semanticweb.org/rycht/ontologies/cyber_security_ontolog
PREFIX platform: <http://www.semanticweb.org/rycht/ontologies/cyber_security_ont
PREFIX cve: <https://cve.mitre.org/about/terminology.html#>
PREFIX oval:<https://oval.mitre.org/language/version5.11/OVAL>

SELECT ?cve ?title ?product ?family1 ?family2
WHERE {
  ?cve a cve:CVE.
  ?cve main:hasTitle ?title.
  ?oval1 a oval:.
  ?oval1 main:hasCVE ?cve.
  ?oval1 main:affectedProduct ?product.
  ?oval1 main:affectedFamily ?family1.
  ?oval2 a oval:.
  ?oval2 main:hasCVE ?cve.
```

```
  ?oval2 main:affectedProduct ?product.
  ?oval2 main:affectedFamily ?family2.
  FILTER (?family1 != ?family2).
}
```

## 7.4 Dáta

postup a popis vyhodnocovania konzistentnosti ABox-u

# Kapitola 8

# Záver

V predchádzajúcej kapitole sme si overili, že naše dáta sú konzistentné a ukázali sme si príklady použiteľnosti našej ontológie. V tejto záverečnej kapitole si zhrnieme, čo sme spravili, aký celkový prínos má naša práca pre publikovanie sémantických spravodajských dát o bezpečnostných hrozbách, ako aj možné rozšírenia našej ontológie.

## 8.1 Zhrnutie

Co sme dosiahli, ako je nase riesenie splnene v porovnani s CTI modelom

## 8.2 Prínos

co sme dosiahli v praci

## 8.3 Budúce pokračovanie

navrhy na vylepsenia

# 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.

[Lab17] UMBC Ebiquity Lab. Uco2. `https://github.com/Ebiquity/uco2`, 2017.

[MB17] Vasileios Mavroeidis and Siri Bromander. Cyber threat intelligence model: An evaluation of taxonomies, sharing standards, and ontologies within cyber threat intelligence. In Joel Brynielsson, 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