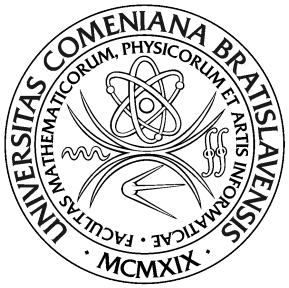
UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

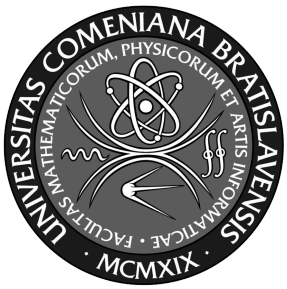


SÉMANTICKÉ PUBLIKOVANIE SPRAVODAJSKÝCH DÁT O BEZPEČNOSTNÝCH HROZBÁCH

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.

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

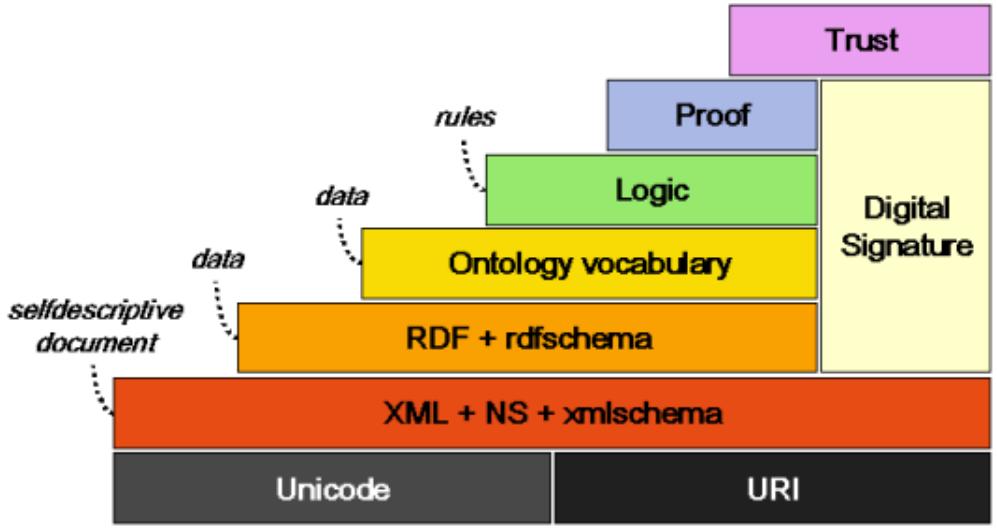
Kapitola 2

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

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

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 zdielať štrukturované dáta na webe pomocou HTML. Časť vízie do budúcna 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.

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...

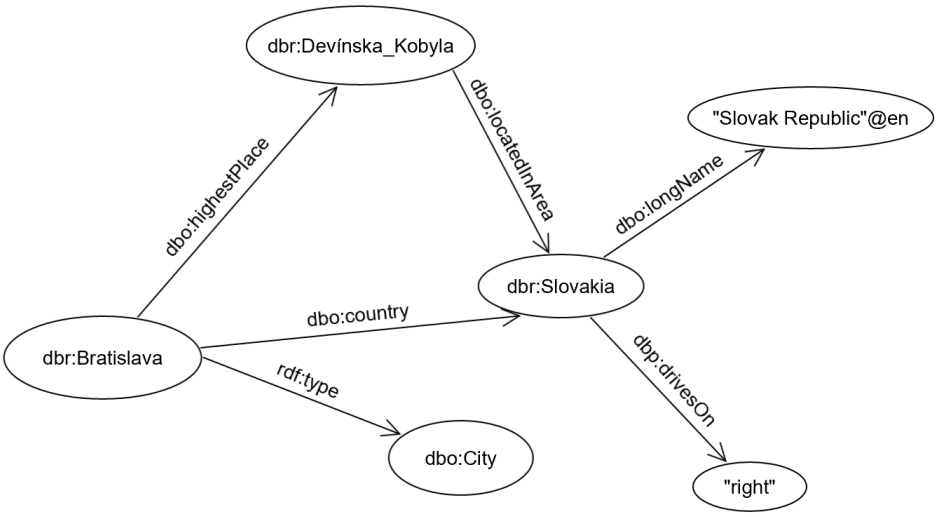
MH: ↓Zvysok ide dobrym smerom, je to presne to, co by som si predstavoval, ze tu budes pisat, len by som to chcel vidiet mozno trochu pomenej, podrobnejsie, systmatickejsie prebrate... Na vacsom priestore, mozno postupne ten priklad budovat... Vysvetlit na nom vsetky zakladne moznosti RDF

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

Web Ontology Language alebo OWL

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)

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. 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.

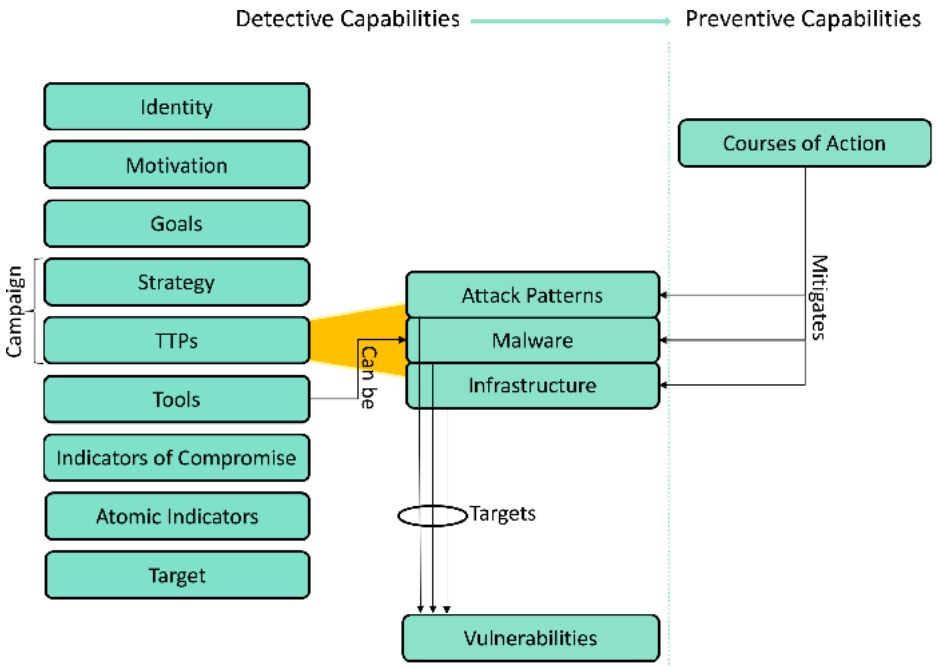
Vďaka URI reprezentácií jednotlivých entít, ktoré sú používané ako identifikátory jednotlivých objektov, nemôže nastať problém nepochopenia dát ako k tomu môže doochádzať v ľudskej reči. Napríklad ak by sme

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.



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

4.1 CTI model

4.1.1 Identita

4.1.2 Útok

4.1.3 Kampaň

Stratégia

TTP

4.1.4 Slabina

4.1.5 Indikátor

4.1.6 Nástroj

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

Návrh riešenia

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

V tejto časti popíšeme ontológiu[Lab17], ktorú sme si vybrali pre ďalší vývoj a budeme porovnávať 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

MH: Ta terminologia, Utok, Indikator, Slabiny, to bude jasne z prech. kapitol?
MR: budem sa presnejsie odkazovat na CTI ktore bude popisane v casti CTI 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.

MH: ↑ "vdaka slovniku OWL" je tazko rozumitelne (skor "pomocou jazyka OWL"): Lepsie by ale bolo, keby tieto veci su v uvodnych prehladovych castiach dostocne vysvetlene natlolo, ze tu uz len staci povedat, ze sa vytvori ekvivalentna trieda
MR: mam pripravene casti o OWL v kapitole syntax ontologii kde neskor doplnim aj syntax (asi manchester) ktoru pouzijeme na niektore upravy a zapis niektorych casti ontologie do prace

Identita je v ontológii UCO reprezentovaná triedou *Attacker*, ďalej útočník, ktorý je namapovaný na existujúcu triedu *ThreatActor* zo STIX-u. 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 systémou 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 určite by to bolo vhodné zadefinovať.

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.

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*

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.

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: ↑ Opat sa stracam, co je Indicator?

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.

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 *affectProduct*, ktoré môžu byť buď hardvérové alebo softvérové a každá slabina ovplyvňuje 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, ktorý tieto dáta uchováva.

Taktiež existuje trieda *CWE*, ktorá je prepojená s triedou *Weakness* z už existujúcej CWE databázy. Taktiež ako trieda *Vulnerability*, popisuje

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.

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 rozsak, 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*).

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.

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.

Kapitola 6

Pravidlá

pravidlá definované v ontológii

Kapitola 7

Dáta

ake data sme importovali, z kadial, akou metodou, kde su ulozene ako sa nad nimi dopytovat, ake odpovede dostaneme – priklady s jazykom SPARQL

Časť III

Testovanie

Konzistetnosť

7.1 Ontológia

postup a popis vyhodnocovania konzistentnosti TBox-u

7.2 Dáta

postup a popis vyhodnocovania konzistentnosti ABox-u

Časť IV

Výsledky práce

Zhrnutie

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

Záver

co sme dosiahli v praci, 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