Optimalizácia prehľadávania subdomén

Samuel Revúcky

doc. RNDr. Martin Stanek, PhD.

Enumerácia (prehľadávanie)

- Dôležitá súčasť obranných stratégií, ale aj útokov
  - Prehľad o infraštruktúre
  - Odhalenie zabudnutých aplikácií, systémov
- Typy metód
  - Pasívne – CT logy, Passive DNS, webové vyhľadávače
  - Aktívne – bruteforce, slovníkový útok
- Nástroje (Amass, Subbrute, ...)

Doména: **uniba.sk**

www.uniba.sk
mail.uniba.sk
ftp.uniba.sk
smtp.uniba.sk
...

Enumerácia (prehľadávanie)

- Čo skúmame?
- Aké predpoklady využívame?
- V čom sa líši náš prístup?

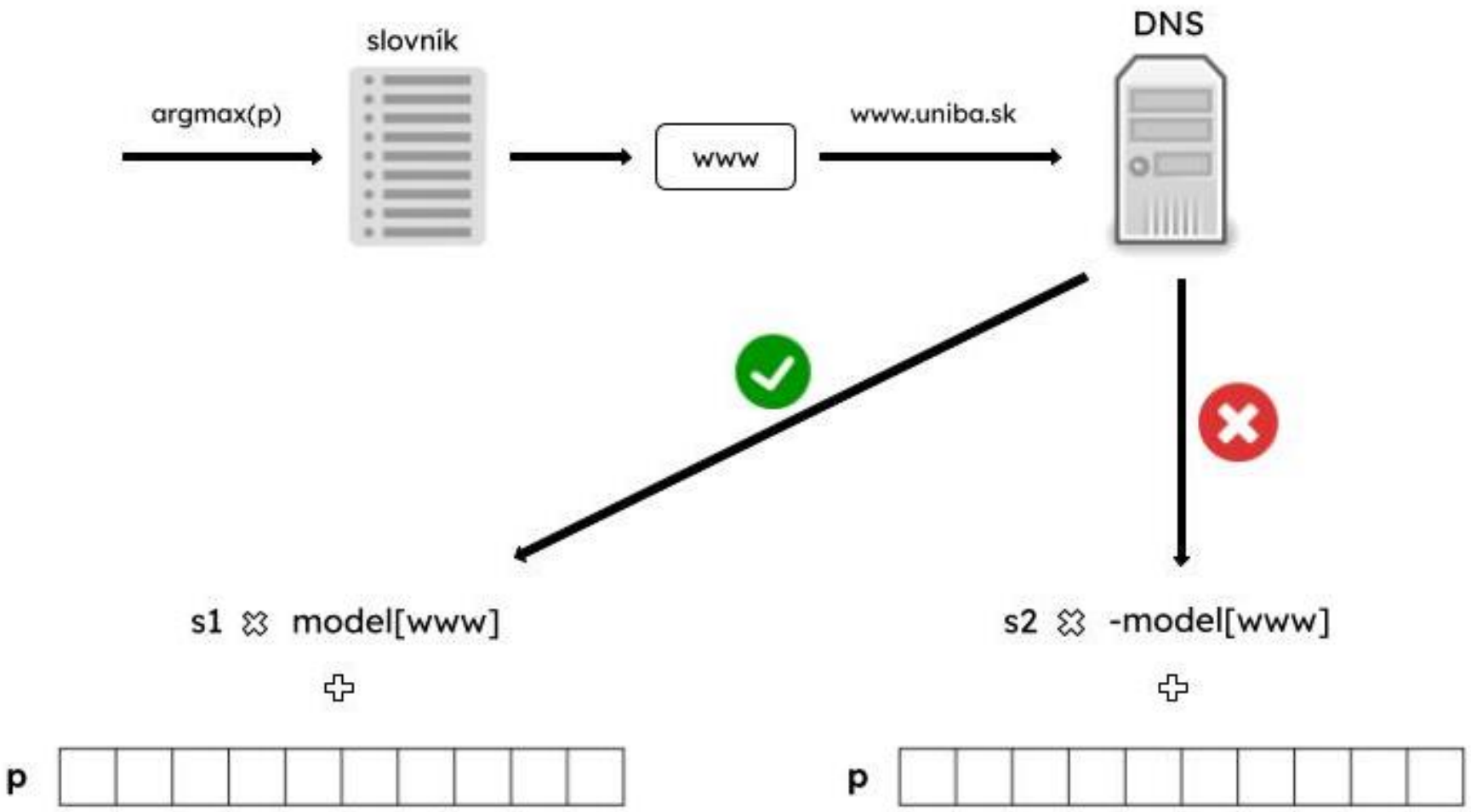
Závislosť medzi menami
cpanel, cpcalendars, cpcontacts, mail, ...

```
X509v3 Subject Alternative Name:
    DNS:ifyouknowyouknow.co.za
    DNS:cpanel.ifyouknowyouknow.co.za
    DNS:cpcalendars.ifyouknowyouknow.co.za
    DNS:cpcontacts.ifyouknowyouknow.co.za
    DNS:mail.ifyouknowyouknow.co.za
    DNS:webdisk.ifyouknowyouknow.co.za
    DNS:webmail.ifyouknowyouknow.co.za
    DNS:www.ifyouknowyouknow.co.za
```

Model závislostí

- Plní potrebu zachytenia vzťahov medzi doménami
- Získanie dát – SAN, logy, crt.sh
- Náhodný výber 10 000 000 ct z roku 2022
- Prienik so slovníkom poskytnutým na enumeráciu
  - Jeden zo štandardných slovníkov Amassu, frekvenčne utriedený podľa výskytu mien
- Reprezentácia závislostí pomocou matice

Algoritmus na enumeráciu



Optimalizácia s1 a s2

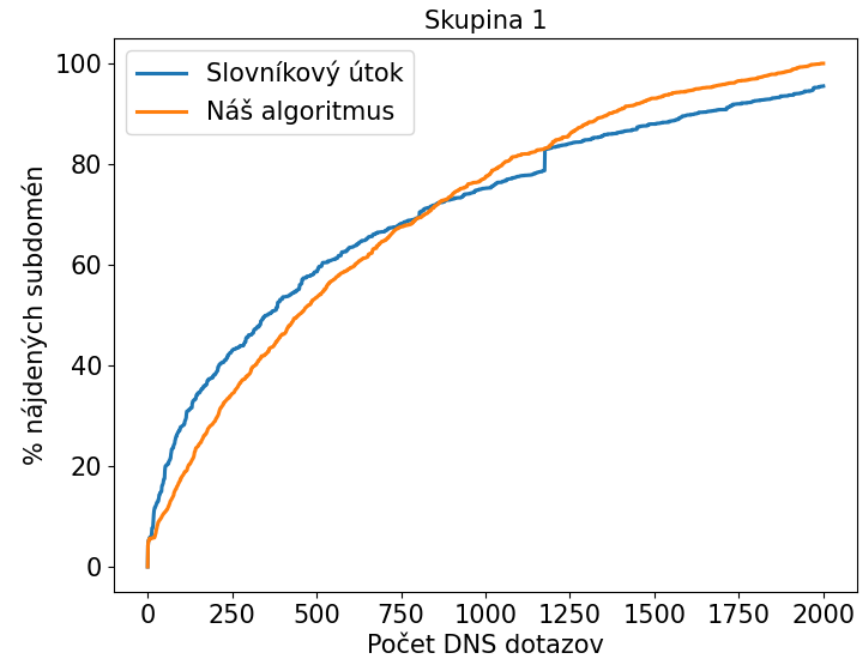
- 50 domén
- 200 DNS dotazov

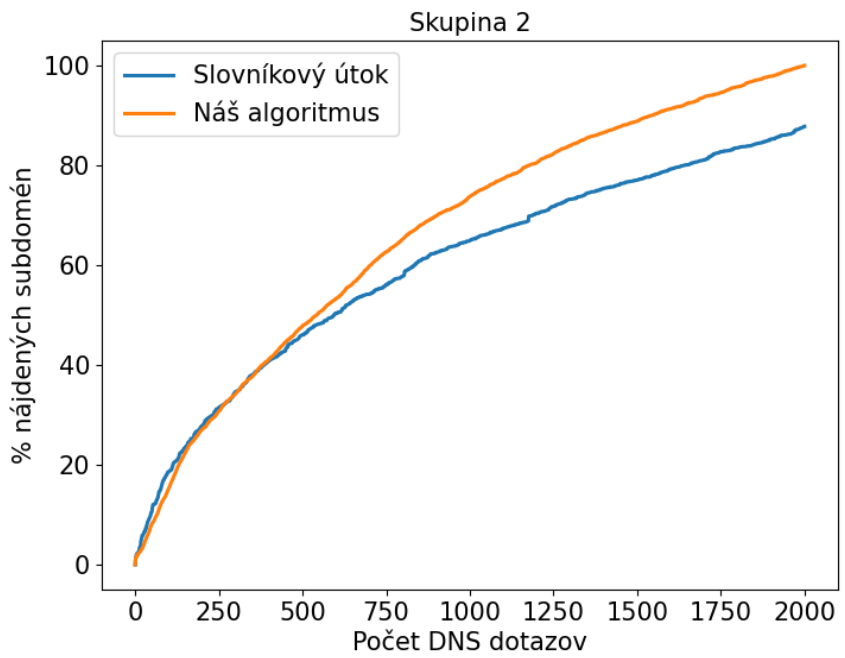
s1 \ s2	7.0	7.5	8.0	8.5	9.0	9.5	10.0	10.5	11.0	11.5	12.0	12.5	13.0
0.5	-0.052	-0.057	-0.019	-0.018	-0.032	-0.013	0.016	-0.006	-0.001	-0.007	-0.031	0.012	-0.019
1.0	-0.075	-0.085	-0.054	-0.051	-0.05	-0.013	-0.058	-0.028	-0.038	-0.049	-0.008	-0.003	-0.005
1.5	-0.045	-0.072	-0.055	-0.064	-0.02	-0.032	-0.005	-0.035	-0.049	0.006	-0.024	-0.009	0.001
2.0	-0.057	-0.05	-0.06	-0.044	-0.01	-0.032	-0.002	-0.032	-0.009	-0.034	-0.021	0.022	0.003
2.5	-0.044	-0.055	-0.047	-0.043	-0.038	-0.026	-0.0	-0.016	0.002	-0.043	-0.015	-0.046	0.005
3.0	-0.053	-0.04	-0.057	0.007	-0.03	-0.021	-0.039	0.009	-0.014	0.003	-0.019	-0.052	-0.034
3.5	-0.055	-0.031	-0.035	-0.042	-0.049	-0.022	-0.033	-0.02	-0.006	-0.038	-0.055	-0.047	-0.035
4.0	-0.021	-0.018	-0.045	-0.039	-0.035	-0.04	-0.004	-0.028	-0.02	0.002	-0.026	-0.043	0.007
4.5	-0.044	-0.039	-0.071	-0.043	-0.041	-0.029	-0.023	-0.042	-0.022	0.007	0.0	-0.011	0.003
5.0	-0.077	-0.04	-0.069	-0.035	-0.056	-0.048	-0.034	-0.024	-0.035	0.01	-0.011	0.005	0.011
5.5	-0.076	-0.022	-0.061	-0.067	-0.064	-0.028	-0.04	-0.007	-0.038	-0.012	-0.005	-0.005	-0.012
6.0	-0.063	-0.043	-0.059	-0.049	-0.077	-0.027	-0.051	-0.014	-0.015	-0.002	-0.002	0.001	0.015
6.5	-0.035	-0.03	-0.07	-0.051	-0.105	-0.068	-0.07	-0.045	-0.029	-0.067	-0.032	-0.036	-0.033
7.0	-0.032	-0.03	-0.06	-0.05	-0.086	-0.067	-0.061	-0.039	-0.037	-0.063	-0.031	-0.034	-0.053

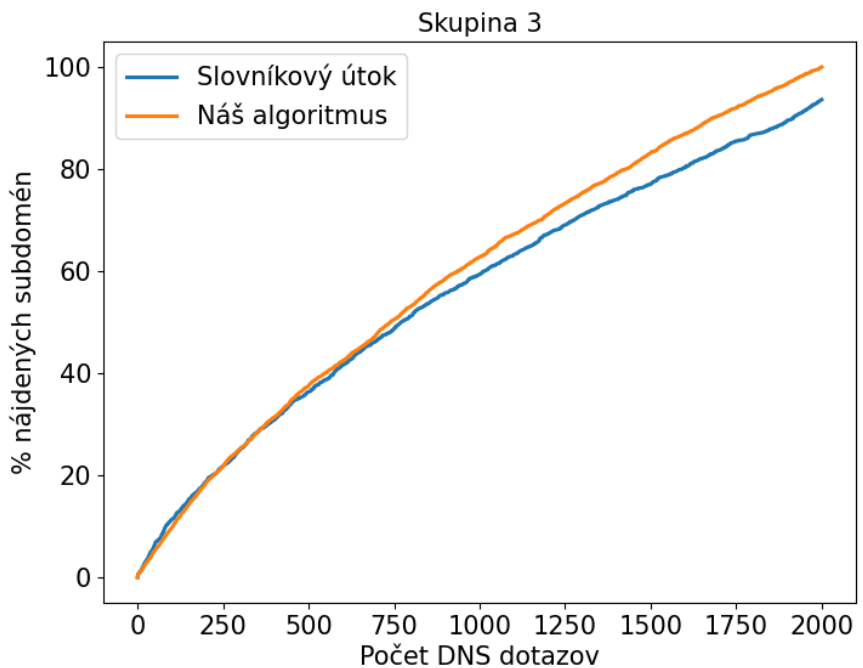
Experimenty

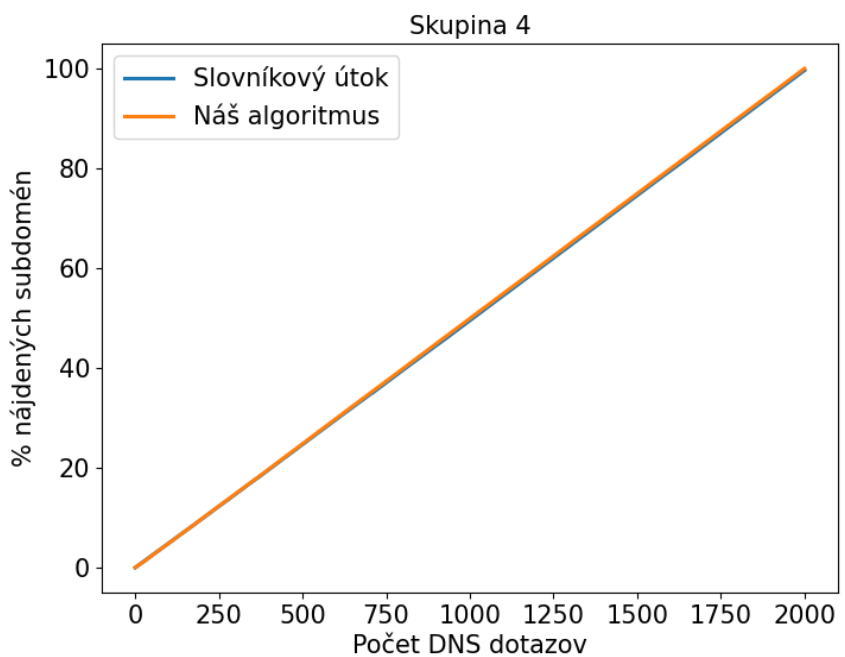
- Porovnanie so slovníkovým útokom
- Nastavenie experimentov
  - 150 domén, 2000 DNS dotazov, s1 = 2,0; s2 = 12,5
- Výsledky
  - Rozdelili sme domény na skupiny

	hustota	počet domén
skupina 1	[0, 100)	65
skupina 2	[100, 300)	52
skupina 3	[300, 1500)	17
skupina 4	[1500,)	16


Skupina 1
Slovníkový útok
Náš algoritmus
% nájdených subdomén
Počet DNS dotazov
0
20
40
60
80
100
0
250
500
750
1000
1250
1500
1750
2000


Skupina 2
Slovníkový útok
Náš algoritmus
% nájdených subdomén
Počet DNS dotazov
0
20
40
60
80
100
0
250
500
750
1000
1250
1500
1750
2000


Skupina 3
Slovníkový útok
Náš algoritmus
% nájdených subdomén
Počet DNS dotazov
0
20
40
60
80
100
0
250
500
750
1000
1250
1500
1750
2000


Skupina 4
Slovníkový útok
Náš algoritmus
% nájdených subdomén
Počet DNS dotazov
0
20
40
60
80
100
0
250
500
750
1000
1250
1500
1750
2000

Záver

- Vytvorili sme model na zachytenie vzťahov medzi doménami
- Pomocou modelu sa nám podarilo výrazne ovplyvniť enumeráciu založenú na slovníkovom útoku
- Využiteľnosť algoritmu pri prehľadávaniach s dobrým počtov dotazov
- Nevhodný pre prehľadávania, kde je čas prioritou a nie je dostupných veľa dotazov
  - Pri 1000 dotazoch bolo trvanie algoritmu približne 110 – 120 sekúnd, oproti asi 40 sekundám pri slovníkovom útoku
- Ďalší vývoj
  - Optimalizácia parametrov
  - Iné CT do modelu, iný zdroj dát

Ďakujem za pozornosť.

Strana 7:

V tabuľke sú certifikáty usporiadané podľa stĺpca id, čo zhruba korešponduje s poradím podľa dátumu platnosti certifikátov.

Certifikáty sú zvyčajne platné viac ako jeden deň. Čo chápete pod dátumom platnosti certifikátu? Myslíte dátum jeho vydania alebo exspirácie, či niečo iné?

crt_pg.py:

```
80: ct_id = random.randrange(first2022, last2022, args.batch_size)
...
85: i += args.batch_size
```

V texte práce píšete: „Prekryvom týchto skupín sme zabránili tým, že sme funkcii dali ako parameter step veľkosť skupiny.“. Ak teda nechcete opakovane sťahovať tie isté dáta, nemôže sa stať že randrange vygeneruje viackrát to isté ct_id (a teda budete zbytočne opakovane sťahovať ten istý blok dát)? K riadku 85: môže sa stať, že select vráti menej záznamov ako je args.batch_size?

- Duplikátov v datasete bolo zhruba 270000, čo tvorilo zhruba 2,8% celého datasetu.

Algoritmus 1:

Prečo vyhadzujete všetky záznamy, kde najdlhší spoločný suffix nemá tvar doména.top_level_doména? Napríklad záznamy, ktoré majú najdlhší spoločný suffix v tvare doména2.domenéna1.top_level_doména, by sa mohli dať použiť.

Strana 12:

Algoritmus sme testovali na dvoch verziách slovníka. Prvou z nich bol vyššie spomenutý subdomains-top1mil-110000.txt a druhou bola jeho verzia frekvenčne utriedená podľa nášho modelu.

Toto tvrdenie sa mi zdá zavádzajúce, lebo ak som ho správne pochopil nejde o dve verzie slovníka, ale o dva rôzne spôsoby inicializácie poľa p s počiatočným rozdelením pravdepodobnosti.

2

Prekvapuje ma konštatovanie, že rovnomerné rozdelenie malo lepšie výsledky ako jeho inicializácia podľa početnosti vo Vašom modeli. Na strane 10 píšete, že „Zjavne ale je už frekvenčne utriedený a zo skúmaných slovníkov má najväčší absolútny prienik s datasetom.“. Preto si myslím, že by bolo zaujímavé vyskúšať ako počiatočné rozdelenie pravdepodobnosti nejakú klesajúcu postupnosť. Tým by sa Váš algoritmus neznevýhodňoval oproti zjavne lepšie frekvenčne utriedenému slovníku.

Strana 16, tabuľka 3.4:

Prečo je slovníkový útok skoro dva krát pomalší ako Váš algoritmus, keď Váš algoritmus je zložitejší? Je to tým, že si v programe odkladáte výsledky predošlých výsledkov DNS dotazov a slovníkový útok sa robí ako prvý? Pretože potom Váš algoritmus, ktorý je v poradí druhý, má výhodu, že veľa dotazov sa načíta z asociatívneho poľa. Akú výpovednú hodnotu majú potom údaje v tabuľke 3.4? Nebolo by pre porovnania výkonových parametrov zaujímavejšie započítať všetky DNS dotazy rovnako obidvom porovnávaným algoritmom?