Detekcia nebezpečných aktivít v záznamoch udalostí mobilných zariadení

Štefan Dlugolinský, Giang Nguyen a Ladislav Hluchý

Ústav informatiky, Slovenská akadémia vied
Dúbravská cesta 9, 845 07 Bratislava

{stefan.dlugolinsky, giang.ui, ladislav.hluchy}@savba.sk

Abstrakt. V článku prezentujeme experiment, ktorý sme vykonali v rámci projektu venujúcemu sa bezpečnosti mobilných zariadení. V experimente sme sa pokúsili aplikovať metódu modelovania jazyka pomocou n-gramov na doménu bezpečnosti mobilných zariadení. Cieľom bolo zistiť, či je možné použiť n-gram modely vytvorené zo záznamov udalostí mobilných zariadení na odhaľovanie nebezpečných udalostí a reťazcov udalostí.

Typ príspevku: Výskumný príspevok

Kľúčové slová: bezpečnosť, n-gramy, modelovanie jazyka

1 Úvod

Predmetom našich experimentov bolo preskúmanie využitia metód pravdepodobnostného modelovania jazyka (v angl. literatúre ako Probabilistic Language Modelling) v doméne bezpečnosti mobilných zariadení. V experimentoch sme namiesto postupnosti slov jazyka modelovali postupnosti udalostí zachytených v záznamoch mobilných zariadení (telefóny a tablety so systémom Android). Tak ako v prirodzenom jazyku, tak aj v záznamoch udalostí sme predpokladali určitú závislosť nasledujúcej udalosti od predchádzajúcich udalostí. V prirodzenom jazyku táto vlastnosť predstavuje sémantiku, kde určitá postupnosť slov dáva nejaký význam. V záznamoch udalostí sme predpokladali, že to je časť nejakého procesu pozostávajúceho z určitých akcií, ktoré sú zaznamenané ako sled udalostí. Našou snahou bolo vytvoriť modely nebezpečných reťazcov udalostí zo záznamov udalostí a využiť vytvorené modely na detekciu podozrivej aktivity v mobilných zariadeniach. V experimente sme aplikovali modely na vzorky udalostí s podozrivou aktivitou ako aj bez nej. Výsledky sme vyhodnotili metrikami perplexity a logaritmickej pravdepodobnosti (pravdepodobnosť, s akou sa testovaná vzorka podobá na vzorky z trénovacej množiny modelu).

2 Prehľad súčasného stavu

Podľa našich zistení sme nenašli literatúru, ktorá by sa zaoberala využitím metód pravdepodobnostného modelovania jazyka na odhaľovanie podozrivých reťazcov zo záznamov udalostí mobilných zariadení. Podobné prístupy však možno nájsť v práci [1], kde autori testovali presnosť detekcie podozrivých častí kódu pomocou n-gram modelov. Predbežné výsledky ukazovali 98% presnosť detekcie pri 3-násobnej krížovej validácii na datasete pozostávanjúcom zo 65 podozrivých programov získaných z emailovej komunikácie. V ďalšej príbuznej práci [3] sa autori zaoberali využitím n-gram modelov na rozpoznávanie neznámych malware v súvislosti s metódou signature-based detection. Ich výsledky ukázali, že n-gram modely dokážu detekovať aj neznáme vzorky kódu.

3 Modelovanie prirodzeného jazyka

Pravdepodobnostné modelovanie jazyka je známe z oblasti spracovania prirodzeného jazyka. Často sa využíva na riešenie rôznych úloh ako napríklad: a) detekcia jazyka, b) automatická korekcia chýb v texte, c) predikcia pri písaní textu, d) strojový preklad a e) rozpoznávanie písaného textu. Princíp spočíva vo vytvorení pravdepodobnostného modelu, ktorý reprezentuje rozdelenie pravdepodobnosti všetkých možných reťazcov slov daného jazyka. Na základe rozdelenia pravdepodobnosti je možné určiť mieru príslušnosti vstupného textu k namodelovanému jazyku, pričom sa berú do úvahy závislosti po sebe idúcich slov v reťazcoch. Tradične sa na modelovanie jazyka používa metóda n-gramov, teda n-tíc slov, ktoré možno pravidlami modelovaného jazyka vytvoriť. N-gramy sa na vytvorenie modelu získajú z trénovacieho textu/trénovacej množiny. Pod pojmom model jazyka rozumieme rozdelenie pravdepodobnosti nad reťazcami trénovacej množiny, pričom model vyjadruje pravdepodobnosť s akou vstupný reťazec predstavuje vetu modelovaného jazyka. Napr., pravdepodobnosť reťazca r dĺžky d pozostávajúceho zo slov $r_1 r_2 \dots r_d = r_1^d = r$ môžeme vyjadriť pomocou vzťahu (1).

$$P(r) = \prod_{i=1}^{d} P(r_i | r_1 \dots r_{i-1}) \tag{1}$$

Výhodnejšie je pravdepodobnosť aproximovať tak, že pravdepodobnosť nasledujúceho slova závisí od slova alebo reťazca slov pred ním. Podľa toho stupňujeme aj modely. Napr. bi-gram model aproximuje pravdepodobosť nasledujúceho slova na základe predchádzajúceho slova; vzťah (2). Tri-gram model zasa na základe dvojice predchádzajúcich slov; vzťah (3). Analogicky takto aproximujeme pravdepodobnosť aj pre modely vyššieho stupňa.

$$P(r) \approx \prod_{i=1}^{d} P(r_i | r_{i-1}) \tag{2}$$

$$P(r) \approx \prod_{i=1}^{d} P(r_i | r_{i-2} r_{i-1}) \tag{3}$$

Dôležitým krokom pri vytváraní n-gram modelu je odhad pravdepodobnosti pre známe n-gramy trénovacieho datasetu. Najjednoduchším spôsobom je odhad pomocou frekvencie n-gramov v datasete (v angl. literatúre Maximum-likelihood estimate). Problémom prístupu n-gramov však je, že najlepšie fungujú vtedy, ak je testovacia množina podobná tej trénovacej. V praxi to však býva niekedy problém. Keďže trénovací dataset

nemôže pokryť všetky možné n-gramy, neznáme n-gramy dostanú pri takomto odhade nulovú pravdepodobnosť – problém riedkeho datasetu. Preto sa môže stať, že ak sa nám neznámy n-gram objaví v testovacom datasete, tak mu model priradí nulovú pravdepodobnosť a vyhodnotí že tento n-gram nepatrí do modelovaného jazyka. Tento problém sa rieši vyhladzovaním odhadu pravdepodobnosti s cieľom priradiť neznámym n-gramom určitú malú pravdepodobnosť. Na vyhladzovanie pravdepodobnosti poznáme niekoľko metód [2]:

a) Add-one smoothing, b) linear interpolation, c) Good-Turing smoothing, d) Jelinek-Mercer smoothing, e) Katz (backoff), f) Witten-Bell smoothing, g) Absolute discounting a h) Kneser-Ney smoothing. Pre potreby našich experimentov sme zvolili algoritmus Knesser-Ney, ktorý sa najlepšie osvedčil v doméne modelovania prirodzeného jazyka. Použili sme pôvodnú verziu s interpoláciou.

4 Dáta

Dáta vo forme záznamov udalostí pochádzali z 18 mobilných zariadení používaných testovacími subjektami a boli zbierané počas obdobia troch mesiacov. Zozbierané záznamy vo forme vektorov s atribútmi sme spracovávali na distribuovanom úložisku tvorenom Hadoop klastrom s nástrojmi Apache Pig a Apache Hive. Zo zariadení sa zbierali dáta o a) uskutočnených hovoroch (CALLS), b) SMS komunikácii (SMS), c) systémových volaniach (INTENT_RECEIVED), d) informáciách o spustených procesoch (PROCESSES), e) sieťovej komunikácii (CONNECTIONS) a f) histórii webového prehliadača (BROWSER_HISTORY).

4.1 Analýza

Zo zozbieraných záznamov sme náhodným výberom vybrali približne 9 mil., ktoré sme podrobili frekvenčnej analýze. Charakteristika vybranej vzorky dát je v Tab. 1.

Tab 1. Frekvenčná analýza vybranej vzorky dát.

typ udalosti	zariadenia	záznamy	unikátne záznamy	
SMS	10	3 231	462	14.30%
CALLS	11	3 187	651	20.43%
INTENT_RECEIVED	18	729 332	572 353	78.48%
PROCESSES	18	4 372 613	3 992 536	91.31%
CONNECTIONS	18	1 951 405	1 797 525	92.11%
BROWSER_HISTORY	15	1 919 961	1 098 979	57.24%
celkovo	18	8 979 729	7 462 506	83.10%

Frekvenčná analýza pozostávala z vytvorenia štatistiky o hodnotách jednotlivých atribútov. Cieľom bolo identifikovať také atribúty, ktoré špecifikujú určitý typ udalosti, aby sme pomocou nich mohli udalosti transformovať na všeobecnejšie slová. Napr. udalostí typu CONNECTION bolo až 92% jedinečných, teda sa skoro vôbec neopakovali. Dôvodom bola široká škála hodnôt atribútov ako aj počet samotných atribútov. Potrebovali sme preto odstrániť niektoré atribúty, prípadne kategorizovať ich hodnoty.

Z udalostí, ktoré obsahovali atribúty: *time*, *gps*, *acc*, *from_addr*, *from_port*, *to_addr*, *to_port*, *state*, *uid*, *application*, *protocol* a *imei* sme ponechali len atribúty *application*, *to_port*, *protocol* a *state*. Tie sme potom transformovali na slová reprezentujúce udalosti v tvare: `connection://{application}/{to_port} /{protocol}/{state}`. Výber atribútov bol urobený na základe odporúčaní experta na zozbierané dáta.

4.2 Predspracovanie

Ako bolo spomenuté v predchádzajúcom príklade, udalosti sme filtrovali a transformovali z vektorov na slová (event), pričom každé slovo malo aj svoju časovú značku (time). Transformácia prebehla podľa nasledovnej schémy:

`time`	`event`
`YYYY-MM-dd HH:mm:ss.SSS`	`type://attr_1/attr_2/.../attr_n`

kde *type* bol typ udalosti (napr. BROWSER_HISTORY pre udalosť z webového prehliadača) a $attr_1$, $attr_2$ až $attr_n$ hodnoty vybraných atribútov (napr. protokol). Udalosti transformované na slová sme ďalej spájali do sekvencií, ktoré boli ekvivalentné vetám v prirodzenom jazyku a udalosti v sekvenciách zasa slovám vo vetách. Udalosti sme spájali tak, aby časový rozdiel medzi dvoma po sebe idúcimi udalosťami v sekvencii nebol väčší ako 10 sekúnd. Túto hodnotu sme zvolili po predchádzajúcej diskusii s expertom na mobilné zariadenia. Spájanie udalostí do sekvencií nám umožňovalo neskôr generovať n-gramy a z nich potom pravdepodobnostný model reťazcov udalostí, podobne ako by to bolo pri texte.

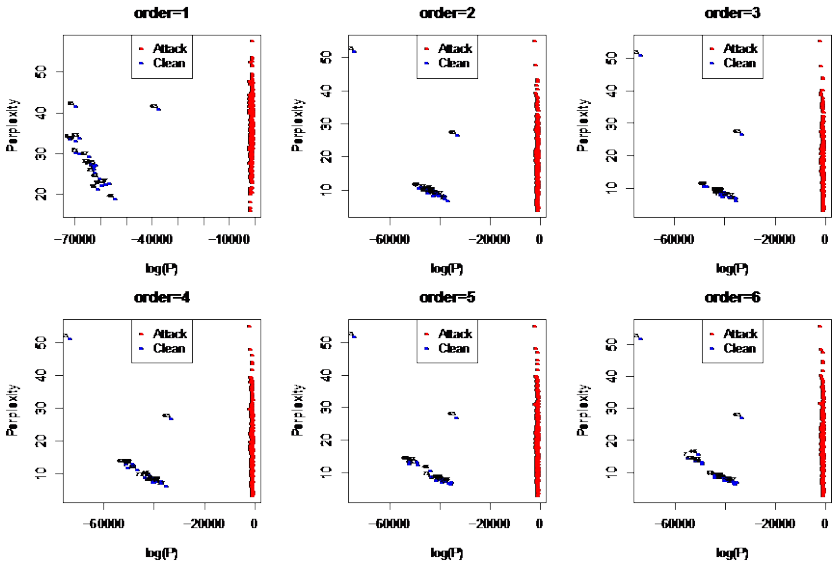
4.3 Vzorky útokov

Pomocou experta boli vykonané simulácie dvoch typov útokov: a) Útok1 - získanie citlivých údajov (Cookies, uložené heslá, auto-fill dáta) a b) Útok2 - získanie vzdialeného prístupu na shell napadnutého zariadenia. K dispozícii sme mali 97 vzoriek útokov typu 1 (59) a 2 (38), ktoré sme zo záznamov získali podľa časových intervalov začiatku a konca simulácie útokov. Vzorky mali v priemere okolo 400 slov (udalostí).

5 Realizácia experimentu

Na odporúčanie experta, ktorý vykonával simulácie útokov, sme si zvolili útok typu 2 a pre každú vzorku útoku tohto typu sme vygenerovali n-gram modely stupňa 1 až 6 s vyhladzovaním pravdepodobnosti algoritmom Kneser-Ney. Pre každý z 38 útokov typu 2 sme tak mali 6 modelov, ktoré sme následne vyhodnocovali nad dvoma typmi datasetov: **CLEAN** Datasety udalostí bez podozrivej aktivity – Vytvorili sme jeden dataset zo záznamov zariadenia A v období štyroch dní. Celkovo obsahoval 47 132 udalostí v 901 sekvenciách. Tento dataset sme používali ako testovací dataset s bezpečnými udalosťami. **ATTACK** Datasety udalostí s útokom typu 2 – Pre každú z 38 vzoriek útoku typu 2 sme vytvorili jeden dataset. Každý dataset bol tvorený sekvenciami udalostí, kde dve udalosti nasledujúce v sekvencii po sebe neboli od seba v čase vzdialené viac ako 10 sekúnd. Z týchto 38 datasetov bolo 19 datasetov zo záznamov zariadenia A. Ostatné boli zo zariadenia B (15 útokov) a z C (4 útoky). Pre potreby experimentu sme si vybrali 19 datasetov zariadenia A, z ktorých sme vygenerovali rovnaký počet n-gram modelov

pre každý jeden stupeň (1 až 6). Tieto modely ďalej spomíname ako *attack* modely. Zvyšné datasety a tiež aj týchto 19 vybraných sme použili ako testovacie datasety s podozrivými udalosťami. Keďže sme testovacie datasety CLEAN a ATTACK vytvorili zo zariadenia, na ktorom boli simulované útoky, overili sme si, že sa nám podozrivé udalosti nedostali do CLEAN datasetu. Evaluáciu *attack* modelov sme vykonali v dvoch krokoch. V prvom kroku sme *attack* modely evaluovali nad všetkými ATTACK datasetmi okrem tých, z ktorých boli modely vytvorené (nezávislosť od trénovacieho datasetu). Vykonali sme tak spolu 342 evaluácií pre každý stupeň n-gramu (1 až 6) a sledovali sme metriky perplexity a logaritmickú pravdepodobnosť príslušnosti reťazcov datasetu k namodelovaným útokom. V druhom kroku sme evaluovali *attack* modely nad CLEAN datasetom. Rovnako ako v prvom kroku, tak aj v druhom kroku sme sledovali metriky perplexity a logaritmickej pravdepodobnosti. Namerané hodnoty získane v oboch krokoch sme zobrazili v grafe na **Obr. 1**. Ako je vidieť z obrázka, hodnoty logaritmickej pravdepodobnosti získané evaluáciou *attack* modelov nad ATTACK datasetmi (krok 1 - červená farba) dosahovali vyššie hodnoty pravdepodobnosti ako pri evaluácii nad CLEAN datasetom (krok 2 - modrá farba; číslami sú podľa vzorky útoku označené *attack* modely). V prípade sledovanej metriky perplexity to už také jednoznačné nebolo. Výsledkom však bolo, že pomocou logaritmickej pravdepodobnosti by bolo možné odlíšiť podozrivé vzorky záznamov udalostí od bežných.



Obr. 1 Výsledky evaluácie attack modelov nad CLEAN a ATTACK datasetmi

6 Záver

Prvotné výsledky vykonaných experimentov ukazujú, že by bolo možné využiť metódy modelovania prirodzeného jazyka aj v oblasti bezpečnosti mobilných zariadení. V experimente sme evaluáciou modelov škodlivých udalostí získali výrazne vyššie hodnoty

logaritmickej pravdepodobnosti pre sekvencie udalostí so škodlivou aktivitou ako bez nej. Toto pozorovanie by sa mohlo využiť napr. na natrénovanie binárneho klasifikátora sekvencií udalostí, ktorý by bol s určitou mierou pravdepodonosti schopný dekekovať nebezpečné aktivity v mobilnom zariadení.

Poďakovanie: Táto publikácia bola podporená projektami VEGA 2/0167/16 a EGI-Engage EU H2020-654142. Zároveň by sme sa chceli poďakovať všetkým kolegom, partnerom a doménovým expertom, ktorí s nami spolupracovali a diskutovali.

Literatúra

1. Abou-Assaleh, T., Cercone, N., Keselj, V., Sweidan, R.: N-gram-based detection of new malicious code. In: Computer Software and Applications Conference, 2004. COMPSAC 2004. Proceedings of the 28th Annual International. vol. 2, pp. 41–42 vol.2 (Sept 2004)
2. Chen, S.F., Goodman, J.: An empirical study of smoothing techniques for language modeling. In: Proceedings of the 34th Annual Meeting on Association for Computational Linguistics. pp. 310–318. ACL '96, Association for Computational Linguistics, Stroudsburg, PA, USA (1996), http://dx.doi.org/10.3115/981863. 981904
3. Santos, I., Penya, Y.K., Devesa, J., Bringas, P.G.: N-grams-based file signatures for malware detection. In: ICEIS 2009 - Proceedings of the 11th International Conference on Enterprise Information Systems, Volume AIDSS, Milan, Italy, May 6-10, 2009. pp. 317–320 (2009)

Annotation:

Detection of malicious activity in the event logs of mobile devices

In this paper, we present experiment conducted within a mobile security project. We tried to apply methods of Natural Language Modelling in the domain of mobile device security. The point was to investigate, whether n-gram models created from event logs of mobile devices can be used to detect malicious events or sequences of such events.