

СИСТЕМА АГРЕГУВАННЯ ТА КЛАСИФІКАЦІЇ ВРАЗЛИВОСТЕЙ В ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ СИСТЕМАХ

Д. А. Савінов^{1, а}, О. М. Барановський^{1, б}

¹ Національний технічний університет України «Київський політехнічний інститут»

Анотація

В даній роботі розглянуто та представлено рішення проблеми агрегації інформації з великої кількості баз даних вразливостей, кожна з яких має власну класифікацію і різне наповнення для одних і тих самих вразливостей. А також створено систему, що агрегує найпопулярніші та найкращі з них та використовує власну загальну класифікацію, що може представити найбільш повно існуючу інформацію про вразливості.

Ключові слова: вразливість, база даних, експлойт, класифікація, text-mining, корпус тексту, вилучення інформації

Вступ

Постійний розвиток інформаційних технологій робить процес захисту інформації все більш складним. Тому організація забезпечення безпеки інформації повинна носити комплексний характер і ґрунтуватися на глибокому аналізі всіх можливих загроз та їх негативних наслідків. При цьому важливо не упустити які-небудь істотні аспекти. Аналіз негативних наслідків припускає обов'язкову ідентифікацію можливих джерел загроз, факторів, що сприяють їх прояву і, як наслідок, визначення актуальних загроз безпеки інформації. У ході такого аналізу необхідно перекоонатися, що всі можливі джерела загроз виявлені, ідентифіковані і зіставлені з всіма можливими факторами (вразливостями), які властиві об'єкту захисту. Тому потрібно мати повну інформацію щодо вразливостей. Для цього існують спеціальні системи їх класифікації. Спеціалізовані бази даних вразливостей не тільки класифікують їх, але й надають можливість вчасно реагувати на їх появу, досліджувати їх більш глибоко, бачити в цілому ситуацію щодо технологій, їх безпечності тощо. Всі ці бази даних вразливостей мають власну класифікацію, а часто і різне наповнення. Тому метою даної роботи є створення системи, що агрегує найпопулярніші та найкращі з них та представляє власну загальну класифікацію, що може представити найбільш повно існуючу інформацію про вразливості.

1. Аналіз інформації в базах даних вразливостей

Сервіси баз даних вразливостей (далі БДВ) – це спеціалізовані сервіси, які накопичують в базах даних структуровану або частково структуровану інформацію про відомі вразливості, експлойти, та класифікують їх або використовують загальні класифікації. Не дивлячись на те, що майже всі системи вико-

ристовують Загальну Систему Оцінки Вразливостей (Common Vulnerability Scoring System, CVSS), яка призначена для класифікації вразливостей за шкалою критичності від 0 до 10, опис вразливостей, надана інформація, власна оцінка є дуже різними. Проаналізувавши існуючі бази вразливостей за критеріями повноти, популярності (тобто рейтингів та відгуків) та оновлюваності було обрано об'єктами для дослідження: exploit-db.com; securitylab.ru; securitytraker.com; securityfocus.com; rapid7.com; cxsecurity; injector (1337day.com); nvd.nist.gov; cve-mitre.org; packetstormsecurity.com.

Для дослідження було обрано 10 різних вразливостей, що відрізнялися за платформою і продуктами, для яких вони існують, вектором атаки, ступенем небезпечності, наявністю рішення та датою знаходження: SolidWorks Workgroup PDM 2014 SP2 – Arbitrary File Write Vulnerability; Adobe Flash Player Integer Underflow Remote Code Execution Exploit; Filemaker Login Bypass and Privilege Escalation Vulnerability; Android Futex Requeue Kernel Exploit; LimeSurvey 2.05+ Multiple Vulnerabilities; VTLIS Virtua InfoStation.cgi – SQL Injection Vulnerability; Foundry CMS Multiple Vulnerability; Unspecified vulnerability in Boosted Boards; Prolink PRN2001 – Multiple Vulnerabilities; SQL Buddy 1.3.3 – Remote Code Execution.

Ці критерії пошуку дають змогу виділити всі можливі поля класифікацій досліджуваних систем, а також проблеми, які повинна вирішувати розроблювана система. З цих полів можна виділити загальні і відмінні. **Загальні:** Name, CVE, Date, Vulnerable apps, versions, Vendor, References, Solution, Platforms, Severity Risk, CVSS, Description, CWE, Tags. **Відмінні за базами:**

- Exploit-db.com: exploit code, Edb-ID, OSVDB-id;
- SecurityTracker.com: SecurityTracker ID, SecurityTracker URL, Message History;
- SecurityFocus.com: BagTraq ID, Class, Discussion, exploit;

^аkeppnew@gmail.com

^бo.baranovskiy@kpi.ua

- Rapid7.com: Actions, Reliability, Development, Module options <code>, Related Modules, Targets;
- NVD: USCert Vulnerability Notes;
- Injector: Tested on.

Для того щоб представляти вразливості у певному вигляді, класифікації, потрібно спочатку отримати самі вразливості, щоб мати з чим працювати. Тому для цього були розроблені програми-парсери, які збирають всю інформацію з представлених вище баз даних вразливостей, а також попередня база даних, що буде містити в собі цю інформацію. Таблиці відповідних баз є об'єктами які є джерелами для загальної класифікації.

Для написання було використано мову Python 2.7 та її мережевий фреймворк Grab::Spider. Перед тим як реалізувати парсери було вивчено структури відповідних систем, що містять обрані для дослідження бази даних вразливостей. Програми, які будуть агрегувати інформацію були поділені на 2 типи:

- 1) Програми, що шукають за заданим словом записи в базах;
- 2) Програми-парсери, що збирають зі знайдених записів всю інформацію, згідно полів в базі.

Таким чином виконується збір інформації за прональзованими базами даних вразливостей.

2. Агрегація інформації

Агрегація різної за своєю структурою інформації з великої кількості баз даних вразливостей є складною проблемою. Тому було створено алгоритми і програмне рішення, які усувають відмінності структурного і класифікаційного плану. Для рішення проблем агрегації інформації розроблений моделі використовується пакет бібліотек для обробки природної мови Natural Language Toolkit (NLTK)[1], який здатний ефективно працювати з англійською мовою, містить велику кількість корпусів текстів, в яких встановлені зв'язки між словами і тому вони можуть бути шаблонами для подальшого визначення зв'язків. До того ж NLTK надає можливість створювати власні корпуси використовуючи машинне навчання. Агрегування інформації для нової класифікації, що проходить за відмінними полями, просто додає інформацію в відповідні поля. Більш складним процесом є агрегування за загальними полями. Такі поля можуть мати інформацію що суттєво відрізняється. Поля ідентифікаторів **CVE** (Common Vulnerabilities and Exposures – загальні вразливості та атаки) та **CWE** (Common Weakness Enumeration – загальний перелік вразливостей) для однієї вразливості ніколи не відрізняються, а тому приводити їх до загального вигляду немає сенсу. Поле **Date** встановлюється тривіальним вибором найпершої дати. Це дозволяє побачити наскільки давно відома ця вразливість. Поля **Vendor**, **Vulnerable apps**, **versions**, **references**, **platforms** встановлюються простим додаванням всієї інформації без дублювання. Проте усунення дублювання також є підзадачею, адже в тексті, що зберігається в базах даних могли бути помилки. Для цього:

- 1) проводиться токенизація (розбиття тексту на слова та словосполучення);
- 2) кожен токен перевіряється за словарем термінів, щоб усунути випадкові помилки;
- 3) почергова сегментація (тобто розбиття на пари рядків, що будуть порівнюватися);
- 4) власне порівняння записів між собою:

- Якщо відстань Хеммінга послідовності токенів буде відмінна за нуль, то вона додається до запису в базу:

$$\sum_{k=1}^p |x_{i_k} - x_{j_k}| = 0 \quad (1)$$

- Якщо ж відрізняється від нуля, то перевіряється той токен в якому були виявлені невідповідності. Перевірка за словником приборала вірогідність того, що була зроблена помилка, тому відмінність є суттєвою і робиться висновок, що це різні рядки.

Найгірше з полем **Description**, адже воно містить інформацію в вигляді природного тексту, тому порівняння цих записів з різних баз даних вразливостей ускладнюється. Тому для вилучення потрібної інформації з різних джерел постають проблеми [2]:

- Co-reference resolution – розв'язання кореференцій;
- Information Extraction – вилучення фактів.

А для поля **Tags** постає ще й проблема **Розпізнавання іменованих сутностей** (Name Entity Recognition). Щоб відкинути вилучення фактів, що не пов'язані з темою потрібно розв'язати **кореференцію** – явище, коли різні посилання ведуть до одного і того ж самого об'єкту. Кореференцією можуть бути:

- **анафора** – посилання на об'єкт за допомогою спеціального слова, терміна, іншого іменування, наприклад: займенники;
- **синонімія** – синоніми, аббревіатури і т.п. [2]

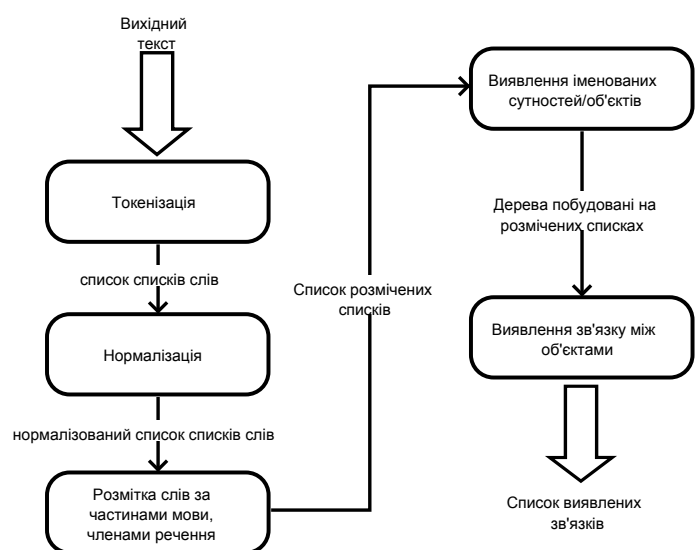


Рис. 1. Модель вилучення фактів

При вилученні фактів з записів текст розділяється на речення та слова, – **токенізується** (tokenization). Наприклад, для однієї з вразливостей текст "Adobe

Reader versions less than 11.2.0 exposes insecure native interfaces to untrusted javascript in a PDF."буде розбито так: ['Adobe', 'Reader', 'versions', 'less', 'than', '11.2.0', 'exposes', 'insecure', 'native', 'interfaces', 'to', 'untrusted', 'javascript', 'in', 'a', 'PDF', '.']

Потім слова **нормалізуються** (chunking, chunking) – виділяється їх початкова форма, частина мови: [('Adobe', 'NNP'), ('Reader', 'NNP'), ('versions', 'NNS'), ('less', 'JJR'), ('than', 'IN'), ('11.2.0', 'CD'), ('exposes', 'NNS'), ('insecure', 'NN'), ('native', 'JJ'), ('interfaces', 'NNS'), ('to', 'TO'), ('untrusted', 'VBN'), ('javascript', 'NN'), ('in', 'IN'), ('a', 'DT'), ('PDF', 'NNP'), ('.', '.'), ('.', '.')]

Далі проводиться частковий синтаксичний розбір і визначаються залежності та зв'язки між словами в реченнях:

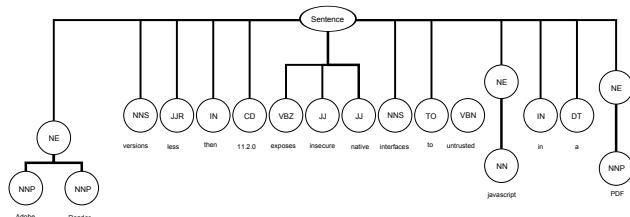


Рис. 2. Дерево речення після розмітки і встановлення іменованої сутності(NE)

Деяку складність становить підзадача вилучення фактів – розпізнавання іменованих сутностей. Її розв'язання дозволить встановлювати мітки(теги, tags) для записів, встановлювати зв'язки для виявлення фактів в текстах. Виявлення іменованих сутностей і ключових слів проходить в декілька етапів:

- 1) вилучення термінів-кандидатів: проходить після токенизації та нормалізації: прибираються стоп-слова, розділові знаки, слова приводяться до початкових форм [1];
- 2) розв'язання кореференцій: відбувається завдяки розмітці членів речення, частин мови і т.п. після порівняння і вибороб значень термінів-кандидатів за словником термінів і списком спеціалізованих статей Вікіпедії – корпусу текстів [3];
- 3) статистична перевірка кандидатів: для кожного терміна-кандидата розраховуються величини:
 - **tf** (term frequency – частота слова) – відношення числа входжень обраного слова до загальної кількості слів документу. Таким чином, оцінюється важливість слова в

межах обраного документу.

$$tf = \frac{n_i}{\sum_{k=1}^p n_k}, \quad (2)$$

де n_i – число входжень терміну в документі, а в знаменнику – загальна кількість слів в документі.

- **idf**(inverse document frequency – обернена частота документу) – інверсія частоти, з якою слово зустрічається в документах колекції. Використання idf зменшує вагу широкочислених слів.

$$idf = \log \frac{|D|}{|d_i \supset t_i|}, \quad (3)$$

де $|D|$ – кількість документів колекції; $|d_i \supset t_i|$ – кількість документів, в яких зустрічається слово (коли $n_i \neq 0$). [4]

- **tf.idf** – показник за яким слова з високою частотою появи в межах документу та низькою частотою вживання в інших документах колекції отримують більшу вагу: $tf.idf = TF \cdot IDF$;

- 4) вилучення прийнятних термінів: $tf.idf$ дає велику вагу іменованим термінам і малу більш загальним поняттям;

Висновки

Розроблено систему класифікації вразливостей в інформаційно-комунікаційних системах на основі агрегування інформації з найбільших спеціалізованих баз даних. Це дозволяє експерту отримати більш точні і повні дані про вразливості, досліджувати їх більш глибоко, бачити в цілому ситуацію щодо технологій, їх безпечності.

Перелік використаних джерел

1. Steven Bird, Ewan Klein, Natural Language Processing with Python – Sebastopol : Safari Books, 2009. – 504 с.
2. Большакова Е.И., Клышинский Э.С. Автоматическая обработка текстов на естественном языке и компьютерная лингвистика: учеб. пособие. – Москва. : МИЭМ, 2011. – 272 с.
3. Information extraction[Електронний ресурс] – Вікіпедія : 2015. – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Information_extraction. –
4. Christopher D. Manning, Foundations of Statistical Natural Language Processing. – Massachusetts : Massachusetts Institute of Technology Press, 2000. – 704 с.