УДК 004.056.53

К. В. ЧУЙКО, В.М. ЧЕШУН
Хмельницький національний університет

МЕТОД АНАЛІЗУ АРГУМЕНТІВ СИСТЕМНИХ ВИКЛИКІВ ДЛЯ ВИЯВЛЕННЯ ЗЛОВМИСНИХ ДІЙ

У роботі з комп'ютером важливим є те, які саме дані і команди вводить користувач. Для забезпечення інформаційної безпеки, актуальною є розробка частини системи, яка буде слідкувати за користувачем.

В статті описано принцип роботи методу захисту, який дозволяє перевіряти команди і виявляти аномальні дії. Розглянуті етапи побудови граматики за допомогою недетермінованого скінченого автомату. Для побудованої граматики використана прихована модель Маркова та обраховано імовірності для вихідних даних. Описані усі важливі моменти і етапи для побудови елементу виявлення аномалій у аргументах системних викликів.

Ключові слова: захист інформації, прихована модель Маркова, недетермінований скінченний автомат

K.V. CHUIKO, V.M. CHESHUN
Khmelnytskyi National University

METHOD OF ANALYSIS OF ARGUMENTS FOR DETECTION SYSTEM CALL MALICIOUS ACTS

In working with a computer is important which data and commands you enter. To ensure information security is important to the development of the system that will keep track of the user.

The paper describes the working principle of this method, which allows the team to check and detect abnormal performance. The stages of construction grammar determined, using a finite automaton. For the constructed grammars using a hidden Markov model and calculated the probability of the source data. Described all the important moments and stages for building element to identify anomalies in the system call arguments.

Keywords: information security, hidden Markov models, Nondeterministic finite automaton

Вступ. Часто прояв дій кібер-зловмисника видно відразу - в системних викликах аргументів незвично довгі рядки або рядки, які містять повторювані недруковані символи. Однак, є ситуації коли зловмисник здатний замаскувати напад таким чином, що всі його запити схожі на регулярні. Наприклад, недруковані символи можуть бути замінені групами друкованих символів [1].

В таких ситуаціях для аналізу потрібна детальна модель аргументу системного виклику. Ця модель може бути набута шляхом аналізу структури аргументу. У такому випадку, структура аргументу - це регулярна граматика, яка описує всі його нормальні, допустимі значення. Структурний висновок - процес, при якому ця граматика виводиться на основі аналізу низки законних рядків, які були визначені під час фази підготовки.

Постановка задачі. Розглянемо підхід до виявлення аномалій, який ґрунтується на аналізі окремих системних викликів. Вхідними даними в процесі виявлення є упорядкований потік $S = \{s_1, s_2, ..., s_n\}$, який складається з системних викликів, що записані операційною системою. Кожний системний виклик $s \in S$ має значення r^s, що повертається, і список аргументів $<a^s_1, ..., a^s_n>$. Відносини між системними викликами або послідовностями викликів, при цьому, не будуть прийняті до уваги. Для кожного системного виклику, який використовується додатком, буде створений власний профіль. Роботу такого принципу, можна розглянути на прикладі додатку, який відправляє повідомлення. Система виявлення вторгнень будує профіль для кожного з системних викликів (таких як відправлення, читання, запис, видалення тощо). Кожен з цих профілів визначає поняття «нормального» стану системних викликів характеризуючи «нормальні» значення для одного або декількох з своїх аргументів. Очікувані «нормальні» значення для окремих аргументів визначаються моделями, що робить актуальною їх розробку.

Виклад основного матеріалу досліджень. Модель являє собою набір процедур, які використовуються для оцінки певного аргументу (наприклад такого, як довжина рядка). Від типу аргументу залежить, які функції можуть бути оцінені моделями. Наприклад, в той час як одна модель працює з описом і розподілом рядкових символів, для роботи з цілими числами її застосувати неможливо.

Система може працювати в одному з двох режимів: режим навчання або виявлення. В режимі навчання система створює моделі і навчається, розробляє поняття «нормальності», створює зразки для подальшого порівняння. Зразки при цьому - значення, які вважаються частиною регулярного виконання програми. Вони або отримані безпосередньо з підмножини вхідних даних S (навчання на льоту) або надані з попередніх станів програми (навчання з набору даних). Важливо, щоб фаза підготовки була вичерпною і без аномальних подій, хоча деякі моделі і мають певний ступінь стійкості до шуму або неповних даних навчання.

Збір якісних даних і їх підготовка є складною проблемою сама по собі.

В режимі виявлення задачею моделі є визначення імовірності виникнення атаки та порівняння значення аргументу на основі даних з попереднього етапу навчання моделі [6]. Перше значення відображає імовірність того, що певне значення функції, яке спостерігається, враховуючи встановлений профіль є аномальним. Для того, щоб класифікувати весь системний виклик як звичайний або аномальний, імовірнісні значення всіх моделей об'єднуються.

Існують два основних припущення для такого підходу.

По-перше, атаки відбуваються завдяки маніпуляціям в аргументах системних викликів. Якщо атака може бути проведена без виконання системних викликів або без впливу на значення аргументів таких викликів, то така техніка не виявить атаку.

По-друге, припущення полягає в тому, що аргументи системного виклику, які використовується при виконанні атаки, суттєво відрізняються від значень, які використовуються під час нормального виконання

додатків. Якщо атака може бути проведена з використанням значення аргументів системних викликів, що не відрізняються від значень, які використовуються під час нормального виконання, то така атака не буде виявлена.

Метою даної моделі є наблизити фактичний, але невідомий розподіл довжин рядкового аргументу і виявити екземпляри, які істотно відрізняються від спостережуваної (нормальної) поведінки. Очевидно, не можна очікувати, що функція щільності імовірності реального розподілу слідуватиме як гладка крива. Можна також припустити, що вона має велику дисперсію. Тим не менш, модель повинна бути в змозі ідентифікувати очевидні відхилення [5].

Розглянемо, наприклад, перший аргумент відкритого системного виклику. Перший елемент рядка символів визначає канонічне ім'я файлу, який повинен бути відкритий. Припустимо, що при нормальній роботі, додаток відкриває тільки файли, що знаходяться в домашньому каталозі додатка і його підкаталогах.

Коли структурний висновок застосовується до аргументу системних викликів, результуюча граматика має бути здатна пройти усі учбові тести. На жаль, немає ніякої унікальної граматики, яка може бути отримана з набору вхідних елементів. Коли ніякі негативні приклади не надані (тобто, елементи, які не мають бути витягнуті з граматики), завжди можливо створити граматику, яка містить точні учбові дані або граматику, яка дозволяє створення довільних рядків. У такому випадку шлях буде відносний і лише у випадку абсолютного шляху варто виявляти підозру [3].

Перший випадок – спрощена форма, оскільки результуюча граматика може отримати тільки навчений вклад, без забезпечення будь-якого рівня абстракції. Це означає, що ніяка нова інформація не виведена.

Другий випадок - форма узагальнення, оскільки граматика здатна до створення усіх можливих рядків, але ніякої структурної інформації не залишається.

Основний підхід, використаний для структурного висновку - узагальнення граматики до тих пір, поки вона вважається прийнятною, і зупинка перед початком втрати занадто великої кількості структурної інформації.

Поняття «прийнятного узагальнення» визначається за допомогою моделей Маркова і імовірності Баєса.

$$P(A) = \sum_{i=1}^{n} P(H_i) * P(A / H_i)$$

де $P(H_i)$ - імовірність гіпотези H_i, $P(A/H_i)$ - умовна імовірність події A при виконанні гіпотези H_i.

На першому етапі розглядаємо набір учбових виробів як продукцію імовірнісної граматики. Імовірнісна граматика - граматика, яка призначає імовірність кожному з його виробів, тобто, деякі слова проводитимуться швидше, ніж інші. Це добре узгоджується із значеннями, зібраними із системних викликів до операційної системи. Деякі значення аргументу звернення до операційної системи з'являються частіше, і представляють важливу інформацію, яка не має бути втрачена в моделюючому кроці.

Імовірнісна регулярна граматика може бути перетворена в недетермінований скінченний автомат (НКА). Кожний стан S автомата має набір з n_s можливих вихідних символів o, які випускаються з імовірністю $p_s(o)$. Кожний перехід t відзначений з імовірністю $p(t)$, яка характеризує імовірність того, що переміщення узяте. Автомат, який зв’язав імовірності, пов'язані з заміщенням символів, і його переходи можна розглядати як модель Маркова [2].

Вихід моделі Маркова складається з усіх шляхів від початкового до кінцевого стану. Значення імовірності може бути призначене для кожного вихідного слова w, тобто отримується послідовність вихідних символів $o_1, o_2, ..., o_k$. Це значення імовірності, як показано у попередній формулі, розраховується як сума імовірностей всіх різних шляхів через автомат, який виробляє w. Імовірність одного шляху є продуктом імовірності генерованих символів $p_{si}(o_i)$ і прийнятих переходів $p(t_i)$.

Імовірності всіх можливих вихідних слів w можна підвести до 1.

$$p(w) = p(o_1, o_2, ..., o_k) = \sum_{p} \prod_{states \in p} p_{s_i}(o_i) * p(t_i)$$

Розглянемо НКА, який зображено на рисунку 1. Імовірності, що пов'язані з кожним переходом ($p(t_i)$), позначені на ребрах графа. Аналогічно, імовірності, що пов'язані з виходом конкретного символу ($p_{si}(o_i)$), наведені на кожному вузлі графа. Щоб обчислити імовірність слова «АВ», потрібно підвести імовірності всіх можливих шляхів, які виробляють цей рядок (в цьому випадку їх два: один, який слідує за лівою стрілкою, і другий, який слідує за правою). Початковий стан не виділяє ніяких символів і має імовірність 1.

$$p(w) = (1.0 * 0.3 * 0.5 * 0.2 * 0.5 * 0.4) + (1.0 * 0.7 * 1.0 * 1.0 * 1.0) = 0.706$$

Мета структурного процесу виведення - знаходження НКА, який має високу імовірність для даних навчальних елементів. Відмінна техніка для отримання моделі Маркова з емпіричних даних полягає у використанні теореми Баєса.

$$p(\text{Модель} \mid \text{Навчальні дані}) = \frac{p(\text{Навчальні дані} \mid \text{Модель}) * p(\text{Модель})}{p(\text{Навчальні дані})}$$

Імовірність навчальних даних вважається масштабованим коефіцієнтом і тому ігнорується. Для досягнення максимального значення імовірності (ліва частина рівняння) нам потрібно максимальне значення, яке показано в правій частині рівняння. Перший термін, який є імовірністю навчальних даних, які отримала модель, може бути вирахуваний для певного автомата додаванням імовірності, вирахуваної для кожного вхідного навчального елементу. Другого терміну, який є попередньою імовірністю моделі, немає як такого. Цим відображається факт, що взагалі менші моделі вважаються кращими [4].

Коли новий аргумент системного виклику аналізується, визначається число входжень кожного символу в рядку. Після цього, значення сортуються в спадному порядку і в поєднанні з шляхом об'єднуються ті значення, які належать до того ж контейнера. Далі застосовується тест для обчислення імовірності того, що даний зразок звертається від ідеалізованого розподілу символів. Отримана величина імовірності p використовується як значення, що повертається для цієї моделі. Коли імовірність того, що

зразок взятий з ідеалізованого символьного поширення збільшується, p збільшує також.

Стандартний тест вимагає наступних кроків, які необхідно виконати:

1) Розрахувати спостережувані і очікувані частоти. Спостережувані значення виводу (одне для кожної групи) вже дані. Очікувана кількість входжень E_i розраховується шляхом множення відносних частот кожного з шести контейнерів (тобто, довжина рядка).

2) Обчислити значення x^2:

$$x^2 = \sum \frac{(o_i - E_i)^2}{E_i}$$

3) Визначити ступені свободи і отримати значення. Ступені свободи для x^2-тесту ідентичні числу доданків у формулі вище, але потрібно відняти одиницю, що дає п'ять замість шести контейнерів. Фактична імовірність p - це зразок отриманий з ідеалізованого розподілу символів (тобто, його значення) і читається з визначеної таблиці за допомогою x^2-значення, яке вважається за індекс.

Імовірнісна модель обчислюється евристично і приймає до уваги загальну кількість N станів, а також число станів t_S і емісії o_S у кожному стані S. Це виправдано тим фактом, що менші моделі можуть бути описані з меншою кількістю станів, а також кількістю викидів і переходів. Фактичне значення може визначатись з формули:

$$p(\text{Модель})\alpha \prod_{S\in states} N^{-(\sum_{S\in states} t_S)} * N^{-(\sum_{S\in states} o_S)}$$

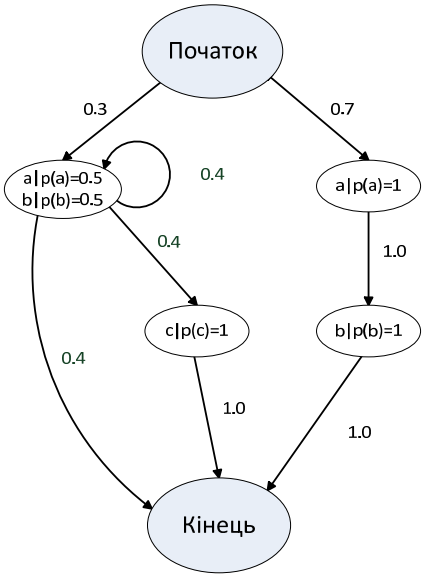

Рис. 1. Приклад моделі Маркова

Значення імовірності моделі, що отримала дані, хронометрує попередні імовірності самої моделі і відображає інтуїтивну ідею, що є конфлікт між простими узагальненими моделями і моделями, які абсолютно відповідають даним, але занадто складні. Моделі, які занадто прості, мають високу зразкову імовірність, але імовірність для створення учбових даних надзвичайно низька. Це призводить до малого значення, коли обидва терміни множаться. Моделі, які занадто складні, мають високу імовірність створення учбових даних (аж до 1, коли модель містить тільки учбовий вклад без будь-яких абстракцій), але імовірність самої моделі безпосередньо дуже низька. Для максимізації значення модель Баєса має наступний підхід: створює автомати з достатнім узагальненням, щоб відображати загальну структуру без відкидання занадто великої кількості інформації.

Висновки

Процес побудови моделі починається з автомата, який точно відображає введення даних, а потім поступово зливає стани. Це злиття станів тривале, імовірність за досвідом далі не збільшується. Після того, як модель Маркова була побудована, може бути використана фаза виявлення оцінки рядкових аргументів. Коли слово є допустимим, вихід моделі Маркова повертає 1. Якщо значення не може бути отримане від даної граматики, модель повертає 0. Такий підхід є перспективний і реалізована на основі нього система буде продуктивною.

Література

1. Шаньгин В. Ф. Защита компьютерной информации. Эффективные методы и средства / Шаньгин В. Ф. – М. : ДМК Пресс, 2010. – 544 с

2. Devarakonda, N.R., Pamidi, S. and Kumari, V.V. (2011); ABIDS system using hidden markov model; Information and Communication Technologies (WICT); 11(14), P. 319–324/

3. D. Wagner and R. Dean. Intrusion detection via static analysis. In Proc. of the 2001 IEEE Symposium on Security and Privacy, pages 156-169, Los Alamitos, CA, May 14-16 2001.

4. Bertacchini, M. and Fierens, P.L.(2007); Preliminary results on masquerader detection using compression based similarity metrics; Electronic Journal of SADIO 2007; 7(1).

5. Ленков С. В. Методы и средства защиты информации : в 2 т. / С. В. Ленков, Д. А. Перегудов, В. А. Хорошко ; под. ред.. В. А. Хорошко. – К, : Арий, 2008. – Т. 2 : Информационная безопасность. – 2008. – 344 с.

6. Лукацкий А. В. Обнаружение атак/ А. В. Лукацкий.-СПб. : БХВ-Петербург, 2003. -256 с.

References

1. Shanhyn V. F. Protection of computer information. Effective methods and tools/ / Shanhyn V. F. – M. : DMK Press, 2010. – 544p.

2. Devarakonda, N.R., Pamidi, S. and Kumari, V.V. (2011); ABIDS system using hidden markov model; Information and Communication Technologies (WICT); 11(14), P.319–324

3. D. Wagner and R. Dean. Intrusion detection via static analysis. In Proc. of the 2001 IEEE Symposium on Security and Privacy, pages 156-169, Los Alamitos, CA, May 14-16 2001.

4. Bertacchini, M. and Fierens, P.L.(2007); Preliminary results on masquerader detection using compression based similarity metrics; Electronic Journal of SADIO 2007; 7(1)

5. Lenkov S. V. Methods and tools for data protection: in 2 vol. / C. V. Lenkov, D. A. Pereghudov, V. A. Khoroshko ; under. ed . V. A. Khoroshko. – K, : Aryj, 2008. – T. 2 : Information Security. – 2008. – 344 s.

6. Lukatskyy A. V. Intrusion Detection / A. V. Lukatskyy.-SPb. : BKhV-Peterburh, 2003. -256 p.

Рецензія/Peer review : 21.12.2014 р. Надрукована/Printed :1.1.2015 р.