ПРИЛОЖЕНИЯ НА КЛЪСТЕРНИЯ АНАЛИЗ В СИСТЕМИТЕ ЗА ОТКРИВАНЕ НА НАРУШЕНИЯ

Веселина Жечева, Евгения Николова
Бургаски свободен университет

CLUSTER ANALYSIS APPLICATIONS INTO ANOMALY-BASED INTRUSION DETECTION SYSTEMS

Veselina Jecheva, Evgeniya Nikolova
Burgas Free Universiti

Анотация: *Системите за откриване на нарушения са важна част от информационната сигурност на съвременните системи. Системите, основани на поведенчески анализ имат ключова роля при откриването на нови, неизвестни атаки или атаки, представляващи варианти на съществуващи атаки. В настоящия доклад се разглеждат някои приложения на клъстерния анализ в системите, основани на поведенчески анализ и са представени някои резултати от симулационни изследвания.*

Ключови думи: *системи за откриване на нарушения, клъстерен анализ, поведенчески анализ*

Abstract: *Intrusion detection systems are essential parts of the contemporary system information security. IDS, based on behavioral analysis have key role in detection of new unknown attacks or variations of existing attacks. The present paper presents some application of the cluster analysis into IDS, based on behavioral analysis and discusses some simulation experiments results.*

Ключови думи: *системи за откриване на нарушения, клъстерен анализ, поведенчески анализ*

1. Увод

Всеки персонален компютър или сървър в съвременния свят е свързан в компютърна мрежа и Интернет и поради това е изложен винаги на риска от неоторизиран достъп от страна на злонамерени лица. Откриването на нарушения (Intrusion Detection) е ключова техника в информационната сигурност, която играе важна роля при откриването на различни видове атаки и осигурява системата. То представлява процесът на наблюдение и анализиране на събитията, произтичащи в наблюдавания компютър или мрежа с цел идентифициране на неправомерни действия от страна на оторизирани потребители или външни лица.

Системите за откриване на нарушения (СОН) изпълняват следните три важни функции: наблюдение, откриване и реагиране на непозволени действия [22]. Те представляват част от цялостна система за информационна сигурност и следят дейността на защитни стени, рутери, управление на сървъри и файлове, критични за други механизми за сигурност. Обикновено СОН предоставят следните услуги:

- Наблюдение и анализ на действията в компютъра и/или мрежата;
- Следене на системната конфигурация и евентуални уязвимости;

- Оценяване на цялостността (интегритета) на критични системни файлове и файлове с данни;
- Търсене на неоторизирани действия.

В съответствие с местоположението си СОН се класифицират на следните категории [4]:

- Хост – базирани – те следят работата на определен компютър (хост), който представлява сървър, предоставящ важни услуги на потребителите. Тези системи записват данни за работата на операционната система и/или важни системни услуги в log-файлове;
- Мрежово – базирани – те следят трафика в определена мрежа или мрежов сегмент. Целта на тези СОН е да открият атаки, предизвикващи отказ на услуга (Denial-of-Service, DoS), сканиране на портове и пасивни атаки, свързани с наблюдение и подслушване на мрежовия трафик;
- Хибридни – включват характеристики и на двата предходни вида.

В съответствие с метода на откриване на атаките СОН се разделят на следните основни видове [12]:

- Базирани на анализ на аномалии в данните (поведенчески анализ, anomaly-based IDS) – основават се на предварително определени профили на нормалната работа на потребителите в системата през определен период от време. След това СОН сканира текущите данни, отразяващи работата на системата, сравнява ги с дефинираните профили и при наличие на значително отклонение издава сигнал за атака или неоторизирано действие. Основното предимство на този метод е потенциалът за откриване на нови, неизвестни средства за атака или варианти на съществуващи атаки, докато основният му недостатък е сравнително високото ниво на фалшиви сигнали.
- Базирани на злоупотреби (сигнатурен анализ, misuse-based IDS) – подобно на антивирусните програми, те се основават на търсене в текущите данни на образци (сигнатури) на известни средства за атака, предварително съхранени в база от данни. Основното предимство на този метод е високата точност, докато неговият основен недостатък е способността за откриване само на известни средства за атака, чиито сигнатури са включени в базата [16].

В настоящия доклад ще разгледаме СОН, базирани на поведенчески анализ. Сред основните проблеми при създаване на такава система е изборът на нивото, на което се описват данните за нормалната работа на системата и се търсят отклонения в тях при сравняването им с текущите данни за работа на системата. Това ниво трябва да бъде достатъчно устойчиво във времето, за да бъдат описани точно потребителските действия и достатъчно гъвкаво, за да реагира надеждно при неоторизирани и отклоняващи се от нормалните профили действия. Работата на тези СОН критично зависи от описанието на нормалните потребителски действия, които могат да бъдат достатъчно сложни и разнообразни, особено при голям брой потребители, което е най-често срещаният случай. Алтернативен метод е предложен от Ko, Fink и Levitt [14], които предлагат описанието на нормалните действия на потребителите да се замени с описание на специален език за описание на спецификациите на действията, извършвани в даден хост от определени процеси, изпълнявани от т.нар. привилегировани процеси, т.е. системни услуги. Тези процеси се изпълняват с по-високи права на достъп до системни ресурси и поради това представляват особен интерес за атакуващите. Освен това тези процеси изпълняват относително постоянни и повтарящи се във времето действия, поради което нормалната им работа може да бъде сравнително лесно дефинирана в профилите на системата. Опростен вариант на този метод е описаният от Forrest [7, 8],

който предлага описанието на нормалната работа се основава на последователности от системни извиквания, наречени n-грами, извършени от тези процеси в рамките на определен период от време. Тези последователности формират базата данни, с която се сравняват данните за нормалната работа на системата.

За реализиране на процеса на откриване на нарушения са предлагани разнообразни методи: крайни автомати [23], скрити Марковски модели [2], машинно обучение [9], data mining [17] и много други. Всички тези методи, наречени наблюдавани (supervised) [10] обаче изискват да бъдат събрани и класифицирани данните, описващи нормалната работа на системата през определен период от време, като през него системата работи незащитена. Този проблем може да бъде решен чрез прилагане на т.нар ненаблюдаван метод, при който не е необходима предварителна информация за нормалната работа на системата. Вместо това СОН извършва наблюдение на текущите данни в системата и търси модел в тях. Всички данни, които се отклоняват от него, се считат за резултат от неоторизирани или злонамерени действия [15]. Описаният подход е реализиран чрез прилагане на клъстерен анализ, при който данните се класифицират като нормални или резултат от неоторизирани действия.

2. Класификационни модели

K-значно клъстеризиране е алгоритъм за клъстерен анализ, който обединява всички обекти в *K* несвързани клъстери, базирани на функцията разстояние. В разглеждания случай целта е да се извърши двоична класификация на данните, т.е. да се разделят на два класа, един, състоящ се от нормалните данни, а от другият – от аномалиите. Алгоритъмът в този случай се състои от следните стъпки:

- Избират се два произволни различни обекти за центрове – един от нормалните наблюдения, а другият – от аномалиите.
- Когато всички наблюдения са класифицирани в най-близките си клъстери, преизчисляват се центровете на клъстерите. *j-тия* нов център се намира по формулата

$$\xi_j = \arg\min_{\xi} \sum_{i:\pi_i=j} d(x_i, \xi),$$

където $\pi_i = \arg\min_j d(x_i, \xi_j)$, $d(,)$ – мярката на разстоянието между два вектора.

За определяне разстоянието между два вектора са използвани следните метрики:

- ***Модел 1.*** *Разстояние на Вагнер – Фишер* [21]. То определя минималния брой операции (вмъкване, изтриване, заместване), които трябва да се извършат с единия вектор, за да стане той идентичен с втория. Това разстояние се задава от формулата:

$$d_{WF}(i,j) = \min \begin{Bmatrix} d(i-1,j) + w(x_i, \varepsilon), d(i, j-1) + w(\varepsilon, y_j), \\ d(i-1, j-1) + w(x_i, y_j) \end{Bmatrix}.$$

където $w(a,b)$ е цената на заместването на елемента *a* с елемента *b*; $w(a,\varepsilon)$ е цената на изтриването на елемента *a* и $w(\varepsilon,b)$ е цената на вмъкването на елемента *b*. Този алгоритъм изисква време от порядъка на O(*mn*) и памет $(m+1)\times(n+1)$, необходима за съхранението на двумерен масив от числа с плаваща точка, където *m* и *n* са дължините на двата вектора.

- ***Модел 2.*** *Разстояние на Жаро* [13]. То се задава по следния начин:

$$d_J = \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right)$$

където m е броят на общите елементи на двата вектора, t е броят на транспозициите, т.е. броят на общите елементи, разделен на 2.

Стъпка 2 се повтаря, докато се намери точният център на всеки клъстер.

3. Симулационни експерименти

Описаната методология е изследвана чрез симулационни експерименти, извършени върху данни, генерирани от [20]. Данните са генерирани в резултат на наблюдение на Unix система в продължение на определен период от време и съдържат описание на работата на привилегировани процеси, които поради естеството на работата си се изпълняват с по-високи от нормалните потребителски права. Поради това те представляват специален интерес за атакуващите. Методите за генериране на данните са описани в [7] и [8], като е обосновано, че те представляват надежден разграничител на нормално и злонамерено поведение. Всеки образец представлява последователност от системни извиквания, които са резултат от изследваните процеси. Данните представляват последователност от наредени двойки числа, като първото число във всяка двойка е идентификаторът на процеса (PID), а второто е номерът на системното извикване. Таблица 1 съдържа някои примери от данните:

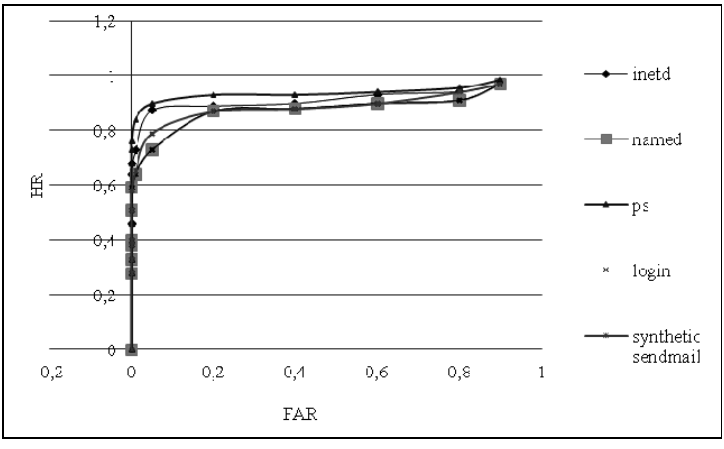
PID	1393	1393	...	1423
System calls	112	19	...	105

Таблица 1. *Данни за системните извиквания*

4. Ефективност на предложената методология

СОН реализират двоична класификация, т.е. определя се дали дадена последователност от наблюдения принадлежи към една от двете групи – множество от нормални дейности на системата или множество от отклонения от нормалната дейност (нарушения). За всяка възможна стойност тестът, чрез който се реализира тази класификация, може да допуска два типа грешки – грешка от първи род (*false positive* - *FP*) и грешка от втори род (*false negative* - *FN*). *FP* се допуска, когато едно събитие се отчита като нарушение, но всъщност това е нормална дейност, докато *FN* е грешката, която се допуска, когато настъпва нарушение, но то не е класифицирано като такова. При двоичната класификация са възможни четири резултата: *FP, FN* и *TP* (*true positive*) и *TN* (*true negative*) - броят на коректно класифицирани нормална дейност и нарушения съответно. Оценката на ефективността и настройката на СОН се нуждае от баланс между тези четирите стойности. Най-често като показатели за измерване на точността на класификацията се използват *ROC кривата и F_1 мярката*.

ROC кривата (*Receiver Operating Characteristic Curve*) [6] представя графично отношението на *степен на попадение* и *относителна стойност на фалшиви сигнали* при различни прагови стойности, т.е. онагледява класификационната способност на СОН. Колкото графиката е по-близо до горния ляв ъгъл, с *попадение* 100% и 0% *относителна стойност на фалшиви сигнали*, толкова по-добра е разпознавателната способност на СОН. Следователно, *ROC кривата* показва цялостно ефективността на класификационните способности на даден тест. За СОН с перфектна класификационна способност графиката на *ROC крива* преминава през горния ляв ъгъл. На Фигура 1 са представени графиките на *ROC криви* за процесите named, inetd, ps, login и synthetic sendmail за модел 1. От графиките на фигурата се вижда, че площта под *ROC кривите* за процесите е между 0,80 и 0,95, което означава, че този метод дава добри резултати при класификация. Аналогични резултати се получават за модел 2 – площта под *ROC кривите* за процесите е между 0,85 и 0,96.



Фигура 1. ROC кривата за процесите named, inetd, ps, login и synthetic sendmail

F_1 е мярка за точност на класификация, която обобщава мерките *Precision* и *Recall* в единствен индикатор [10] и се определя по формулата

$$F_1 = 2.\frac{Precision.Recall}{Precision + Recall},$$

където

$$Precision = \frac{TP}{TP + FP},$$
$$Recall = \frac{TP}{TP + FN}.$$

Високата ѝ стойност гарантира, че и двете мерки *Precision* и *Recall* показват добри за класификацията стойности. Таблица 2 съдържа получените стойности на F_1 – мярката за изследваните данни.

Процес	$d_{WF}(i,j)$		d_J	
	RI	***F*$_1$-measure**	***RI***	***F*$_1$-measure**
synthetic sendmail	0,5724	0,614	0,7724	0,689
ps	0,8836	0,883	0,8796	0,822
login	0,7021	0,875	0,7321	0,863
named	0,7384	0,861	0,7482	0,869
inetd	0,6331	0,677	0,7341	0,699

Таблица 2. Стойности на RI и F_1 – мярката за процесите named, inetd, ps, login и synthetic sendmail за двата модела

Една от мерките за качеството на клъстерните алгоритъми е индексът на Ранд (Rand index – *RI*) [18]. *RI* изчислява колко подобни са клъстерите, получени от алгори-

тъма за групиране. На този индекс може да се гледа като на мярка за процента на правилните решения, взети от алгоритъма. Той се изчислява по следната формула

$$RI = \frac{TP + TN}{TP + TN + FN + TN}.$$

Един от проблемите по отношение на *RI* е, че *FP* и *FN* имат еднакво тегло. *RI* приема стойности между 0 и 1, където 1 означава, че двата получени клъстера са много еднакви. В Таблица 2 са представени стойностите на *RI* за процесите named, inetd, ps, login и synthetic sendmail за двата изследвани модела. Както може да се види от стойностите, посочени в таблицата, те са все още доста големи. Това показва, че броят на сходства от клъстерни двойки е малко и качеството на класификацията на този метод е по-скоро добра.

При класифициране трябва да се анализират структурите на клъстерите относно размера и разстоянието между клъстърите. За целта се въвеждат две разстояния:

- Вътреклъстърно разстояние (Intra-cluster distance) – размера или компактността на всеки клъстър. То се пресмята като се използва разстоянието като диаметър, като осреднен диаметър или като диаметър относно центъра [1].
- Междуклъстърно разстояние (Inter-cluster distance) – разстоянието между клъстърите, което се пресмята по един от следните начини:
- Като единична връзка, която е най-близкото разстояние между две наблюдения, принадлежащи на два различни клъстъра;
- Като цялостна връзка, която е най-отдаленото разстояние между две наблюдения, принадлежащи на два различни клъстъра;
- Като осреднена връзка, която е средно разстояние между всички наблюдения, принадлежащи на два различни клъстъра;
- Като централна връзка, която е разстояние между центровете на два различни клъстъра.

Някои от методите на валидност, чрез които се оценява компактността на клъстърите и разстоянията между тях са:

1. Индекс на валидност на Дейвис-Болдин [3], който се изчислява по формулата

$$DB(K) = \frac{1}{k}\sum_{i=1}^{k} \max_{i \neq j} \left\{ \frac{\Delta(K_i) + \Delta(K_j)}{\delta(K_i, K_j)} \right\}.$$

Той приема ниска стойност, ако клъстърите са компактни и далеч един от друг, т.е. ниската стойност показва добра клъстеризация.

2. Профилен индекс [19]. Ширина на профила на *i*-тото наблюдение от *j*-тия клъстър се изчислява по формулата

$$s_i^j = \frac{x_i^j - y_i^j}{\max\left\{x_i^j, y_i^j\right\}}.$$

От израза се вижда, че $-1 \le s_i^j \le 1$. Чрез него се дефинира профил на клъстър K_j

$$S_j = \frac{1}{m_j}\sum_{i=1}^{m_j} s_i^j.$$

Глобалният профилен индекс за клъстеризация се дава чрез

$$S = \frac{1}{n}\sum_{j=1}^{n} S_j \ .$$

Лесно се вижда, че профилът на клъстъра и глобалния профилен индекс приемат стойности между -1 и 1.

3. Индекс на Дън [5] се дефинира като частно на минималното междуклъстърно разстояние и максималното вътреклъстърно разстояние

$$D = \frac{\delta_{\min}}{\Delta_{\max}} \ .$$

Този индекс се ограничава в интервала $[0,\infty)$ и трябва да се максимизира.

4. *C*-индекс [11] се дефинира като

$$C = \frac{S - S_{\min}}{S_{\max} - S_{\min}} \ ,$$

където S е сума от разстоянията на всички двойки наблюдения от един и същи клъстър, m е броят на тези двойки, $S_{\min}$ е сума на m най-малки разстояния, ако се разгледат всички двойки наблюдения и $S_{\max}$ е сума на най-големите разстояния по всички двойки. Този индекс е ограничен в интервала $[0,1]$ и трябва да се минимизира.

Високата хомогенност в клъстерите и високата разнородност между клъстерите показват, че е постигнато добро групиране. За да се оцени валидността на методите ние използваме индекс на валидност на Дейвис-Болдин, Индекс на Дън и С-индекс, чиито стойности за нашите експерименти са дадени в таблица 3.

Таблица 3. Стойности на индекс на валидност на Дейвис-Болдин, Индекс на Дън и С-индекс за двата модела

Процеси	**Индекс на валидност на Дейвис-Болдин**	**Индекс на Дън**	**С-индекс**
synthetic sendmail	3,921	1,733	0,3487
ps	3,935	1,742	0,3291
named	3,584	1,696	0,4498
login	3,590	1,690	0,4475
inetd	3,587	1,685	0,4491

а/ за модел 1

Процеси	**Индекс на валидност на Дейвис-Болдин**	**Индекс на Дън**	**С-индекс**
synthetic sendmail	3,827	1,773	0,4837
ps	3,865	1,728	0,3946
named	3,694	1,715	0,4507
login	3,731	1,702	0,4601
inetd	3,656	1,713	0,4394

б/ за модел 2

5. Заключение

В настоящия доклад бяха представени изследвания на приложението на клъстерния анализ в системите за откриване на нарушения, базирани на аномалии, като чрез разделянето на данните в два клъстера е реализирана двоичната класификация. Анализът на резултатите от извършените симулационни изследвания показва ефективността на изследваната методология. Бъдещите изследвания включват приложение на други методи от клъстерния анализ в СОН и сравнение на получените резултати.

Литература:

1. Николова Е., В. Жечева, Приложение на техники от клъстерния анализ в системите за откриване на нарушения, сп. „Компютърни науки и комуникации", бр.1/2012, стр.42-47, БСУ, Бургас, 2012.
2. Bhole A.T., A.I.Patil, Intrusion Detection with Hidden Markov Model and WEKA Tool, International Journal of Computer Applications (0975 – 8887), Vol. 85, No 13, January 2014, pp. 27-30.
3. Bolshakova N. and Azuaje F., Cluster Validation Techniques for Genome Expression Data, Signal Processing, 83, 2003, pp. 825-833.
4. Butun, I.; Morgera, S.D.; Sankar, R., A Survey of Intrusion Detection Systems in Wireless Sensor Networks, IEEE Communications Surveys & Tutorials, (Volume:16, Issue: 1), 2014, pp. 266 – 282.
5. Dunn, 1974. Dunn, J. (1974) Well separated clusters and optimal fuzzy partitions, Journal of Cybernetics, 4, 95-104.
6. Ferri C., N. Lachinche, S. A. Macskassy, A. Rakotomamonjy, eds., *Second Workshop on ROC Analysis in ML*, 2005.
7. Forrest S., S.A. Hofmeyr, A. Somayaji, T.A. Longtaff, A Sense of Self for Unix Processes. In Proceedings of the 1996 IEEE Symposium on Security and Privacy, IEEE Computer Society Press, Los Alamitors, CA, pp.120-128.
8. Forrest S., S.A. Hofmeyr, A. Somayaji, Intrusion detection using sequences of system calls, Journal of Computer Security Vol. 6, 1998, pp. 151-180.
9. George A., Anomaly Detection based on Machine Learning: Dimensionality Reduction using PCA and Classification using SVM', International Journal of Computer Applications (0975 – 8887) Volume 47– No.21, June 2012.
10. Görnitz N., M. Kloft, K. Rieck, U. Brefeld, Toward Supervised Anomaly Detection, Journal of Artificial Intelligence Research, Vol. 46 (2013), pp. 235-262.
11. Hubert L, Schultz J. Quadratic assignment as a general data-analysis strategy. British Journal of Mathematical and Statistical Psychologie, 1976; 190-241.
12. Jaiganesh V., S. Mangayarkarasi, Dr. P. Sumathi, Intrusion Detection Systems: A Survey and Analysis of Classification Techniques, International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 4, April 2013, pp. 1629-1635.
13. Jaro M. A., Advances in record linking methodology as applied to the 1985 census of Tampa Florida, Journal of the American Statistical Society, 1989, 414-420.
14. Ko C, G. Fink, K Levitt. Automated Detection of Vulnerabilities in Privileged Programs by Execution Monitoring, Proceedings of the 10th Annual computer Security Applications Conference, pp.134-144, 1994.
15. Mahmood D.Y., M. A. Hussein, Feature Based Unsupervised Intrusion Detection, World Academy of Science, Engineering and Technology International Journal of

Computer, Control, Quantum and Information Engineering, Vol:8, No:9, 2014, pp. 1549-1553.
16. Mitchell R., Ing-Ray Chen, A survey of intrusion detection in wireless network applications, Computer Communications 42 (2014), pp. 1–23.
17. Nadiammai G.V., M. Hemalatha, Effective approach toward Intrusion Detection System using data mining techniques, Egyptian Informatics Journal, Vol. 15, Issue 1, March 2014, Pages 37–50.
18. Rand W. M., Objective criteria for the evaluation of clustering methods, Journal of the American Statistical Association, 66 (336), 846–850, 1971.
19. Rousseeuw, P.J., Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, Journal of Computational and Applied Mathematics, 20, 1987, 53-65.
20. University of New Mexico's Computer Immune Systems Project, http://www.cs.unm.edu/~immsec/systemcalls.htm.
21. Wagner R. A., M. J. Fischer, The string-to-string correction problem, Journal of the Association for Computing Machinery 21, 1974, pp. 168-173.
22. Wang D., Yeung, D.S., and Tsang, E.C., „Weighted Mahalanobis Distance Kernels for Support Vector Machines", IEEE Transactions on Neural Networks, Vol. 18, No. 5, Pp. 1453-1462, 2007.
23. Xu Y.; J. Jiang; R. Wei; Y. Song, TFA: A Tunable Finite Automaton for Pattern Matching in Network Intrusion Detection Systems, IEEE Journal on Selected Areas in Communications, Vol.:32 , Issue: 10, 2014, pp. 1810 – 1821.

За контакти:

Веселина Жечева, доцент, д-р, БСУ, тел. 900-487, vessi@bfu.bg
Евгения Николова, доцент, д-р, БСУ, тел. 900-413, enikolova@bfu.bg